

Alternating Minimization Algorithm with Automatic Relevance Determination for Transmission Tomography under Poisson Noise

Yan Kaganovsky*, Shaobo Han*, Soysal Degirmenci†, David G. Politte‡, David J. Brady*, Joseph A. O’Sullivan‡, and Lawrence Carin*. §

Abstract. We propose a globally convergent alternating minimization (AM) algorithm for image reconstruction in transmission tomography, which extends automatic relevance determination (ARD) to Poisson noise models with Beer’s law. The algorithm promotes solutions that are sparse in the pixel/voxel-differences domain by introducing additional latent variables, one for each pixel/voxel, and then learning these variables from the data using a hierarchical Bayesian model. Importantly, the proposed AM algorithm is *free of any tuning parameters* with image quality comparable to standard penalized likelihood methods. Our algorithm exploits optimization transfer principles which reduce the problem into parallel 1D optimization tasks (one for each pixel/voxel), making the algorithm feasible for large-scale problems; This approach considerably reduces the computational bottleneck of ARD associated with the posterior variances. Positivity constraints inherent in transmission tomography problems are also enforced. We demonstrate the performance of the proposed algorithm for x-ray computed tomography using synthetic and real-world datasets. The algorithm is shown to have much better performance than prior ARD algorithms based on approximate Gaussian noise models, even for high photon flux. Sample code is available at <http://www.yan-kaganovsky.com/#!/code/c24bp>.

Key words. Alternating minimization, Automatic relevance determination, Transmission tomography, Poisson noise, X-ray CT, Optimization Transfer.

1. Introduction. Tomographic image reconstruction is the process of estimating an object from measurements of its line integrals along different angles [1]. Tomographic reconstruction is an ill-posed problem [2] in the sense that there are multiple solutions that are consistent with the data. It is therefore desirable to make use of prior knowledge, which includes the statistical characteristics of the measured data, and properties that are expected of the image.

The images in transmission tomography represent attenuation per unit length of the impinging beam due to energy absorption inside the imaged object. Well known examples are x-ray computed tomography (CT) [3], and electron tomography [4]. The attenuation must be nonnegative, since a negative value corresponds to an increase in the intensity of the exiting beam, which is non-physical. These images can often be well approximated by a sparse representation in some transform domain. In some applications, sparsity is present directly in the native image domain [5], but more commonly it is present in the pixel-difference domain [6] or in some transform domain, such as the wavelet, curvelet or shearlet transforms [7, 8].

There are generally two types of approaches for tomographic image reconstruction. The first type consists of one-shot algorithms, such as filtered back-projection [1] and its extensions, which rely on analytic formulas. These algorithms do not incorporate the type of prior knowledge mentioned above and also produce prominent artifacts when some of the data are

§Emails: yankagan@gmail.com, shaobohan@gmail.com, s.degirmenci@wustl.edu, dgpolitte@gmail.com, dbrady@duke.edu, jodyosullivan@gmail.com, lcarin@duke.edu.

*Department of Electrical and Computer Engineering, Duke University, Durham, NC 27708, USA.

†Department of Electrical and Systems Engineering, Washington University in St. Louis, MO 63130, USA.

‡Mallinckrodt Institute of Radiology, Washington University in St. Louis, MO 63110, USA.

missing. The second type consists of iterative algorithms which are based on minimizing some objective function. The latter enable the incorporation of prior knowledge about the data and the image of interest, e.g., the objective function can put less emphasis on measurements that are expected to be more noisy, include penalties that promote sparsity of the image in some representation, and positivity constraints for the image can also be included. In addition, iterative algorithms are far more robust to missing data.

Many transmission tomography problems involve physical processes consisting of quanta, e.g., the number of transmitted photons in x-ray CT, or the number of transmitted electrons in electron tomography. Therefore, there is a strong physical motivation for modeling these processes using independent Poisson random variables [4, 6]. In the medical CT community, the Poisson noise model is commonly used [6] and provides the basis for a well known class of iterative algorithms [6, 9, 11, 12, 13]. In addition, the mean of each measurement is modeled according to Beer’s law [6], which is an inherent non-linearity in transmission data.

In this paper, we develop a new class of statistical iterative algorithms for image reconstruction in transmission tomography that is inspired by automatic relevance determination (ARD) [14, 15, 16, 17]. It incorporates prior knowledge that includes Poisson statistics of the underlying signals, Beer’s law, positivity of the image, and also sparsity of the image in some underlying representation. What sets this new class apart from prior art is the fact that it *automatically learns the balance between data-fidelity and prior knowledge*, thus avoiding the use of any tuning parameters and the difficulty in determining them. In addition, it also computes posterior variances, which can be used for adaptive sensing/experimental design, or for determining the system’s sensitivity to noise. To the best of our knowledge, the proposed algorithm is the first ARD-based algorithm for Poisson noise that is both scalable to very large problems and that is also guaranteed to converge.

Our work is motivated by x-ray CT which is used in medical and security applications [1]. Nevertheless, the methods presented here can be applied more generally to any type of transmission tomography that involves count data. In formulating the problem presented in Sec. 1.1 we have tried to retain some generality in the hope that other imaging modalities could potentially benefit from the proposed approach. Naturally, this entails some simplifications of the forward model which are discussed later in Sec. 7.

It should be noted that most modern commercial CT scanners do not employ photon counting detectors. However, there are already photon counting detectors used in clinical studies [18], and it is certainly possible that they will be integrated into commercial CT scanners in the near future. Modern CT scanners also utilize a high photon flux, which led various researchers to propose algorithms that are based on an approximate post-log Gaussian noise model [19, 20, 21] rather than a Poisson noise model. The Gaussian model can be derived by making a 2nd order Taylor series approximation to the log-likelihood corresponding to the Poisson noise model with Beer’s law, under the assumption of moderate to high photon flux. This results in a data-weighted least-squares datafit term [19], which also corresponds to a post-log Gaussian noise model. However, in low-count scenarios, which are critical to maintain low radiation dose in pediatric and small animal CT, there could be a substantial performance benefit in using the Poisson model once photon-counting detectors become widely available. For example, it has been shown empirically that the approximate Gaussian model leads to a systematic negative bias that increases as photon flux decreases [22].

1.1. Model. We consider a Poisson noise model for transmission tomography [6] given by

$$(1.1) \quad p(\mathbf{y}|\mathbf{x}) = \prod_{i=1}^n p(y_i|\mathbf{x}) = \prod_{i=1}^n \text{Poisson}[\eta_i \exp(-\boldsymbol{\phi}_i^T \mathbf{x})],$$

where here and henceforth $p(\mathbf{y}|\mathbf{x})$ denotes the likelihood function, $\mathbf{y} \in \mathbb{Z}^n$ contains the measurements, $\text{Poisson}[\lambda]$ denotes the Poisson univariate probability distribution with rate λ , and the exponential function encompasses Beer's law (a physical property of the signals [6]). The column vector $\mathbf{x} \in \mathbb{R}_+^{p \times 1}$ denotes the lexicographical ordering of the 2D/3D image to be reconstructed, where the entry x_j is the linear attenuation coefficient at the j th pixel/voxel in units of inverse length. The operation $\boldsymbol{\phi}_i^T \mathbf{x}$ represents a line integral over the image corresponding to the i th source-detector pair (see Fig. 1) and to the measurement y_i . We denote by $\boldsymbol{\Phi} \in \mathbb{R}_+^{n \times p}$ the system matrix that is formed by concatenating the row vectors $\boldsymbol{\phi}_i^T \in \mathbb{R}_+^{1 \times p}$ ($i = 1, 2, \dots, n$). η_i is the mean number of photons that would have been measured using the i th source-detector pair if the imaged object was removed. We make the standard assumption that η_i was measured for all i in an independent calibration experiment and is known to us.

We construct $\boldsymbol{\Phi}$ using the approach in [23], where an entry ϕ_{ij} at the i th row and j th column of $\boldsymbol{\Phi}$ is equal to the length at which the line between the i th source-detector pair intersects the j th pixel/voxel (see Fig. 1). Each ray transverses at most $O(p^{1/D})$ pixels/voxels, where p is the total number of pixels/voxels and $D = 2, 3$ for 2D/3D images, respectively. Accordingly, each row $\boldsymbol{\phi}_i^T$ in the system matrix has only $O(p^{1/D})$ non-zero entries. Henceforth, we absorb a reference linear attenuation coefficient into $\boldsymbol{\Phi}$ so that \mathbf{x} is dimensionless.

1.2. Background. The method proposed in this paper is an extension of automatic relevance determination (ARD). For the convenience of readers unfamiliar with ARD we first provide background to explain the potential advantages of ARD motivating this work and discuss the key differences between ARD and other more common sparse estimation methods. Our aim here is to emphasize previous challenges in adapting ARD to the Poisson model in (1.1), challenges which we aim to resolve with the proposed method.

1.2.1. Maximum a Posteriori (MAP) Estimation. Statistical iterative methods for CT [6, 9, 11, 13] cast the problem of image reconstruction as a penalized maximum likelihood estimation problem, given by

$$(1.2) \quad \mathbf{x}^* = \arg \max_{\mathbf{x}} [\log p(\mathbf{y}|\mathbf{x}) + \log p(\mathbf{x}|\beta)],$$

where $\log p(\mathbf{y}|\mathbf{x})$ is the log-likelihood function which represents the datafit. For the Poisson model in (1.1), maximizing the log-likelihood is equivalent to minimizing the I-divergence¹ between the actual measurements \mathbf{y} and the predicted measurements $\boldsymbol{\eta} \odot \exp(-\boldsymbol{\Phi}\mathbf{x})$ [13], where \odot denotes an elementwise product. $p(\mathbf{x}|\beta)$ in (1.2) is the prior probability distribution representing prior knowledge about \mathbf{x} and $\log p(\mathbf{x}|\beta)$ is a penalty that promotes solutions with desired properties according to this prior knowledge, e.g., the ℓ_1 penalty [5], or the total variation (TV) penalty [25, 26], which promote sparse solutions. β is a tuning parameter

¹I-divergence between two vectors $\mathbf{p}, \mathbf{q} \in \mathbb{R}_+^M$ is defined as $I(\mathbf{p}||\mathbf{q}) = \sum_{j=1}^M p_j \log(\frac{p_j}{q_j}) - p_j + q_j$ [24]

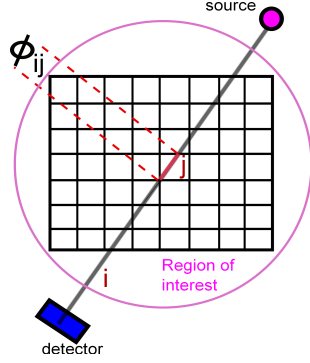


Figure 1. Tomographic measurements in 2D. The total linear attenuation for the i th measurement is equal to the line integral over the imaged domain along the path defined by the i th source-detector pair. Several measurements are collected simultaneously using several detectors. The source and detectors are rotated to collect different views. Each line integral is modeled as $\phi_i^T \mathbf{x} = \sum_j \phi_{ij} x_j$ where x_j is the linear attenuation coefficient at the j th pixel in units of inverse length and ϕ_{ij} is the intersection length of the i th line with the j th pixel. If p is the total number of pixels, then each line intersects at most $O(p^{1/2})$ pixels.

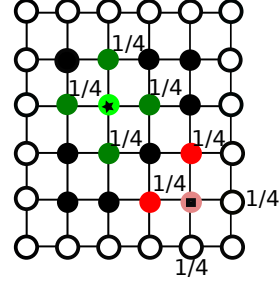


Figure 2. An example for an adjacency graph defining a neighborhood Θ in (2.3). Each node represents a pixel in the image and $\{j, k\} \in \Theta$ iff the j th and k th nodes are connected by an edge. In this example, the node colored in light green and marked by a star has 4 neighboring pixels shown in dark green. Each neighbor has a weight of $1/4$ when computing their average. The image is augmented to introduce a boundary of pixels with zero values, represented by white nodes. The latter are not included in the forward model and are merely introduced to provide the interpretation for the $1/4$ weight when averaging less than 4 neighbors, e.g., consider the node colored in light red and marked by a square.

that determines the tradeoff between the datafit and the prior, or the bias-variance tradeoff. The “best” value of β is usually unknown and depends on the imaged object at hand. To determine β , it is often required to solve (1.2) many times for different values of β , examine all reconstructed images and then select the value that best fits the needs of the application, based on some chosen figure of merit. For many common choices of $p(\mathbf{x}|\beta)$ considered below, decreasing β will put more weight on the datafit term and will result in increased variance and in noisier images. Increasing β will put more weight on the penalty term, and will result in increased bias and increased artifacts in the image, e.g., staircase artifacts. For a given β , the solution in (1.2) can be interpreted as the maximum a posteriori (MAP) solution, i.e., the maximizing point of the posterior distribution, which is given by Bayes’ theorem

$$(1.3) \quad p(\mathbf{x}|\mathbf{y}) = p(\mathbf{y}|\mathbf{x})p(\mathbf{x})/p(\mathbf{y}).$$

The ℓ_1 and TV penalties are non-differentiable and do not allow the use of standard gradient-based methods for solving (1.2). An alternative to TV is the class of neighborhood roughness penalties (Gibbs priors) [11] which have the following form

$$(1.4) \quad -\log p(\mathbf{x}|\beta) = \beta \sum_{\{j,k\} \in \Theta} w_{jk} \pi(x_j - x_k),$$

where w_{jk} is a positive weight inversely proportional to the distance between the centers of the j th and k th pixels/voxels, Θ is a set of ordered pairs $\{j, k\}$ defining a neighborhood system and $\pi(x)$ is chosen to be a positive, even, and twice continuously differentiable function. To

preserve sharp edges in the image, $\pi(x)$ is often chosen such that $\pi(x) \propto |x|$ for large $|x|$ and can be viewed as a smooth version of the ℓ_1 penalty. A common choice is [10, 11]

$$(1.5) \quad \pi(x) = \delta^2[|x/\delta| - \log(1 + |x/\delta|)],$$

which approaches the quadratic x^2 for $\delta \rightarrow \infty$ and the linear function $\delta|x|$ as $\delta \rightarrow 0$. This type of penalties allows the use of standard gradient-based methods but comes with the price of having an additional tuning parameter δ , so the search for the tuning parameters becomes even more cumbersome. Decreasing δ too much will result in noisy images but with sharp edges, and increasing δ too much will result in blurred images.

1.2.2. Automatic Relevance Determination (ARD). A different approach to sparse estimation is “automatic relevance determination” (ARD) [14]. Originally, ARD has been proposed for *Gaussian* likelihoods with the prior [15]

$$(1.6) \quad p(\mathbf{x}|\boldsymbol{\gamma}) = \mathcal{N}(\mathbf{x}|\mathbf{0}, \boldsymbol{\Gamma}), \quad \boldsymbol{\Gamma} = \text{Diag}(\boldsymbol{\gamma}),$$

where $\text{Diag}(\boldsymbol{\gamma})$ denotes a diagonal matrix with main diagonal $\boldsymbol{\gamma}$. In this approach sparsity is promoted by treating \mathbf{x} as latent variables and seeking the newly introduced *hyperparameters* $\boldsymbol{\gamma} \in \mathbb{R}_+^p$ that maximize the marginal likelihood $p(\mathbf{y}|\boldsymbol{\gamma})$ (also called the “model evidence” [16])

$$(1.7) \quad \boldsymbol{\gamma}^* = \max_{\boldsymbol{\gamma}} \log p(\mathbf{y}|\boldsymbol{\gamma}).$$

Once $\boldsymbol{\gamma}^*$ is found, the estimate for \mathbf{x} is given by the mean of the posterior distribution $p(\mathbf{x}|\mathbf{y}, \boldsymbol{\gamma}^*)$. Interestingly, the solutions to (1.7) have many $\gamma_j \approx 0$ and the posterior distribution for the corresponding parameters x_j becomes highly concentrated around zero². The mechanism that promotes sparsity of $\boldsymbol{\gamma}^*$ has been revealed by Wipf et al. [27, 28], with the analysis specifically tailored for Gaussian likelihoods.

The solution of (1.7) was originally addressed by Tipping in [15] using an expectation-maximization (EM) algorithm [29]. In the E-step, the Gaussian posterior probability distribution $p(\mathbf{x}|\mathbf{y}, \boldsymbol{\gamma})$ is computed for given hyperparameters $\boldsymbol{\gamma}$. In the M-step, $\boldsymbol{\gamma}$ are updated based on the mean and variances of the posterior distribution found in the E-step. As opposed to MAP estimation methods, which focus on the mode (maximum) of the posterior distribution, ARD methods also compute the covariance matrix (during the E step), or at least the marginal variances, which provide additional information. ARD-based algorithms are therefore computationally more demanding than MAP methods, mainly due to the expensive task of estimating the variances. Previous extensions of ARD [28, 30] require the same type of operations and rely on approximations to the variances [30] to maintain scalability.

Despite computational complications, the ARD approach can be quite rewarding and excel in hard or ill-posed problems, as evidenced in several recent applications [31]. A very significant advantage of ARD methods is their ability to *automatically learn the balance between the datafit and the degree of sparsity*. This avoids the tuning of nuisance parameters, which is often nonintuitive, object-dependent, and requires many trials. The posterior variances are key in determining this tradeoff. The variances can also be used for adaptive sensing/experimental design [17, 30], which is outside the scope of this paper and will be the subject of future work.

²By employing Bayes’ rule in (1.3), it can be easily shown that if the prior is concentrated around $x_i = 0$ then the corresponding posterior distribution is also concentrated around $x_i = 0$

1.2.3. Difficulties in Previous ARD-based Methods for Poisson Models. The ARD-EM algorithm [15] described in Sec. 1.2.2 cannot be directly applied to the Poisson likelihood in (1.1) since the E-step is intractable due to the non-conjugacy³ of the prior in (1.6) and the likelihood. Although we are not aware of any previously published work on ARD for Poisson models with Beer’s law, a simple remedy to the non-conjugacy is the Laplace method, discussed in [15]. In that approach, the posterior is approximated during the E-step using a multivariate Gaussian distribution, with the mean equal to the MAP solution and the covariance matrix equal to the inverse of the Hessian matrix for the log-likelihood, evaluated at the MAP solution. However, the resulting “EM-like” algorithm is not based upon a consistent objective function, *and is not guaranteed to converge*. In addition, it requires inversions of the Hessian matrix *with computational complexity* $O(p^3)$, where p is the total number of pixels/voxels. This is infeasible for x-ray CT, where p could be very large, e.g., $p = 10^8$ or higher. Instead, one can use more sophisticated methods [30, 32, 33] to indirectly estimate the diagonal of the inverse, i.e., the marginal variances, since only these are required for updating the hyperparameters γ [15]. Note however that the work in [30, 32, 33] does not consider Laplace approximations, so these methods have not been tested for this case. Irrespective of how the variances are computed, the “EM-like” algorithm with the Laplace approximation is unsatisfactory from a principled perspective due to the lack of an objective function and due to the lack of convergence guarantees. The objective is also desired in order to assess convergence in practice and in order to verify the correctness of the implementation. *In addition, we are faced with conceptual difficulties, since the analysis that revealed the sparsity promoting mechanism of ARD by Wipf et al. [16, 27, 28] is restricted to Gaussian likelihoods.*

1.2.4. Previous ARD Method for Transmission Tomography Based on an Approximate Post-Log Gaussian Noise Model. As mentioned in the introduction, it is reasonable to replace the model in (1.1) with an approximate post-log Gaussian noise model under moderate to high photon flux conditions. An ARD algorithm for transmission tomography based on the Gaussian model and the EM algorithm in [15] was proposed by Jeon et al. [17]. It should be noted that the original ARD algorithm in [15] has a computational cost of $O(p^3)$ due to the computation of posterior variances, and therefore, it is not practical for large scale problems. In contrast, the algorithm proposed in [17] scales as $O(np^{1/D})$ where $D = 2, 3$ for 2D and 3D images, respectively. There are several disadvantages of the algorithm in [17]: (1) it does not enforce positivity of the solution; (2) due to the approximate estimation of posterior variances, the overall algorithm does not have convergence guarantees; (3) it requires one to choose the type and number of probing vectors for estimating the variances [17], which depends on the specific geometry of the system and on the imaged object; (4) in order to estimate the posterior variances it requires solving *many* $p \times p$ matrix equations, which is computationally expensive; (5) computing the objective function scales as $O(p^3)$, making this operation infeasible for large scale problems; (6) it is limited to proper priors and complete sparse representations.

1.3. Contributions. The contributions of this paper are summarized as follows.

1. We present a new alternating minimization (AM) algorithm for transmission tomogra-

³A prior is said to be conjugate to the likelihood if both the prior and posterior probability distributions are in the same family. The posterior is then given in closed-form.

phy, which extends automatic relevance determination (ARD) to Poisson noise models with Beer’s law. The proposed algorithm is free of any tuning parameters and unlike prior ARD methods, also has a consistent objective function that can be evaluated in a computationally efficient way, in order to assess convergence in practice and to verify the correctness of the implementation.

2. We reveal the sparsity-promoting mechanism in the proposed ARD-based algorithm for Poisson likelihoods. This preserves insights similar to those gained from previous work [16, 27, 28] for Gaussian likelihoods.
3. We derive a *scalable* AM algorithm for transmission tomography, that is well adapted to parallel computing. We reduce the computational complexity considerably by simplifying the algorithm into parallel 1D optimization tasks (for each pixel/voxel) with the scalar mean and variance updates also done in parallel.
4. We present a convergence analysis for the scalable AM algorithm and establish guarantees for global convergence⁴. To the best of our knowledge, this is the first *ARD-based* algorithm for *Poisson* models that has convergence guarantees.
5. We study the performance of the algorithm on synthetic and real-world x-ray datasets.
6. We compare the proposed algorithm to the Gaussian ARD method in [17] and demonstrate that the proposed algorithm leads to better performance under Poisson distributed data, even for high photon flux.

1.4. Outline of the Paper. The rest of this paper is organized as follows. In Sec. 2 we present a general overview of the proposed AM algorithm and explain how sparsity is promoted. In Sec. 3 we present a scalable AM algorithm based on separable surrogates for the objective function, which is well adapted to parallel computing and feasible for large-scale problems. In Sec. 4 we present a detailed analysis of the convergence properties of the algorithm presented in Sec. 3. In Sec. 5 we make connections between the proposed algorithm and related previous methods. We also present an alternative view of the proposed ARD method that further explains how sparsity is promoted. In Sec. 6 we extend the proposed ARD framework to allow the modeling of sparsity in *overcomplete* representations and the use of improper priors, while retaining the convergence properties studied in Sec. 4. In Secs. 7–8 we discuss model limitations and possible alternative optimization methods related to our work. In Sec. 9 we study the performance of the proposed scalable ARD algorithm on synthetic and real-world datasets (with a parallel implementation used in the experiments) and compare it to prior MAP estimation methods as well as to the ARD method in [17] which is based on a post-log Gaussian noise model. Conclusions are presented in Sec. 10.

2. Variational ARD.

2.1. Prior. In many applications of transmission tomography sparsity is not present directly in the native basis but in some transform domain. Therefore, we consider a generalized prior (similar to Seeger and Nickisch [30]) to address these cases

$$(2.1) \quad p(\mathbf{x}|\boldsymbol{\gamma}) = \mathcal{N}(\mathbf{x}|\mathbf{0}, (\boldsymbol{\Psi}^T \boldsymbol{\Gamma}^{-1} \boldsymbol{\Psi})^{-1}), \quad \boldsymbol{\Gamma} = \text{Diag}(\boldsymbol{\gamma}),$$

⁴An algorithm is said to be “globally convergent” if for *arbitrary starting points* the algorithm is guaranteed to converge to a point satisfying the necessary optimality conditions (local optima included) [34].

where $\gamma \in \mathbb{R}_+^p$ are newly introduced hyperparameters, and we assume a transformation

$$(2.2) \quad \mathbf{s} = \Psi \mathbf{x},$$

where Ψ is known and \mathbf{s} is approximately sparse. By choosing $\Psi = \mathbf{I}$, the prior in (2.1) reduces to (1.6). To provide some insight into the mechanism of the proposed algorithm we assume for now that Ψ is square and invertible. We shall present a more general approach in Sec. 6 to allow for cases where these assumptions are not met in practice.

Any scalable algorithm will rely on efficient matrix-vector multiplication $\Psi \mathbf{x}$. This is achieved if Ψ is sparse or has some special structure, such as discrete Fourier or wavelet transforms. In this work we focus on neighborhood penalties, which are commonly used for x-ray CT, the application explored in Sec. 9. We first consider Ψ given by

$$(2.3) \quad (\Psi)_{jk} = \psi_{jk} = \begin{cases} 1 & \text{if } j = k \\ -1/N_j & \text{if } \{j, k\} \in \Theta, \\ 0 & \text{otherwise} \end{cases},$$

where Θ is the set of ordered pairs $\{j, k\}$ defining a neighborhood system and N_j is the number of neighbors for the j th pixel/voxel. An example is shown in Fig. 2. The neighborhood is usually chosen to be small enough such that Ψ will be sparse with $N_j + 1$ non-zero elements in the j th row. For the choice in (2.3), each s_j in (2.2) is equal to the difference between the j th pixel/voxel and the average of its neighboring pixels/voxels, i.e., $s_j = x_j - \sum_{k: \{j, k\} \in \Theta} [x_k / N_j]$. Note that sparsity of these pixel/voxel differences implies piecewise smooth images. Lastly, we note that the choice in (2.3) implies Dirichlet boundary conditions, which is equivalent to assuming additional neighbors outside the image domain with values set to zero, such that the number of neighbors for each pixel/voxel is equal, i.e., $N_j = N$ (see Fig. 2). These boundary conditions agree with physical constraints typical of many real experimental setups, where the attenuation is zero outside the region of interest. We shall remove these restrictions in Sec. 6.

2.2. Variational View of the EM Algorithm for ARD. We start with a known variational view [35] of the EM algorithm which is used to solve (1.7) for Gaussian noise models. Here we introduce some of the notation used throughout this work, and discuss the type of difficulties encountered when trying to extend ARD to non-Gaussian noise models.

We rewrite the model evidence as

$$(2.4) \quad \log p(\mathbf{y}|\gamma) = -\mathcal{F}[q(\mathbf{x}), \gamma] + D_{KL}[q(\mathbf{x})||p(\mathbf{x}|\mathbf{y}, \gamma)],$$

where

$$(2.5) \quad \mathcal{F}[q(\mathbf{x}), \gamma] = \mathbb{E}_{q(\mathbf{x})} \log[q(\mathbf{x})/p(\mathbf{y}, \mathbf{x}|\gamma)],$$

$$(2.6) \quad D_{KL}[q||p] = \mathbb{E}_{q(\mathbf{x})} \log[q(\mathbf{x})/p(\mathbf{x}|\mathbf{y}, \gamma)].$$

$q(\mathbf{x})$ is the proposed variational distribution and $p(\mathbf{x}|\mathbf{y}, \gamma)$ is the exact posterior distribution, \mathcal{F} is the *free variational energy* (**FVE**) or negative evidence lower bound (**ELBO**), $D_{KL}[q||p]$ is the Kullback-Leibler (KL) divergence, and \mathbb{E}_q is the expected value with respect to $q(\mathbf{x})$. The EM algorithm can be viewed [35] as minimizing the free variational energy (FVE) function

\mathcal{F} in (2.5) (or maximizing the ELBO) by alternating between updating $q(\mathbf{x})$ in the E-step and updating γ in the M-step, as summarized below

$$(2.7) \quad \mathbf{E}: \quad q^{(t+1)} = \arg \min_q \mathcal{F}[q(\mathbf{x}), \gamma^{(t)}],$$

$$(2.8) \quad \mathbf{M}: \quad \gamma^{(t+1)} = \arg \min_{\gamma} \mathcal{F}[q^{(t+1)}(\mathbf{x}), \gamma].$$

In the E-step, the optimal solution with zero KL divergence is $q^{(t+1)} = p(\mathbf{x}|\mathbf{y}, \gamma^{(t)})$, i.e., the exact posterior based on the previous $\gamma^{(t)}$ values, which can be computed directly from (1.3). In the M-step, minimizing \mathcal{F} with respect to γ can only increase the KL term, since it is nonnegative and overall the model evidence is also increased.

For *non-Gaussian* likelihoods and the prior in (2.1), the E step in (2.7) is intractable, i.e., the expression for $p(\mathbf{x}|\mathbf{y}, \gamma^{(t)})$ is unknown and its direct calculation according to (1.3) involves a high-dimensional integral that is prohibitive to compute. Instead, we are forced to *restrict* $q(\mathbf{x})$ to a tractable family of distributions to approximate the posterior. However, this results in an algorithm that is no longer guaranteed to increase the evidence [36]. This occurs because the KL term in (2.4) is never zero and can either decrease or increase after updating γ . Furthermore, a local maximum of the evidence with respect to γ is generally *not* a local minimum of \mathcal{F} (or a local maximum of the ELBO), except for degenerate cases [36].

Therefore, when considering the above approach to extend ARD, we immediately encounter a *conceptual* difficulty, since most previous theoretical studies of ARD [16, 27, 28] are based on the *maximization of the evidence* and its specific closed-form expression for the Gaussian likelihood. It is not immediately clear how to extend the ARD framework to the Poisson noise model in (1.1) in a manner which will preserve the insights gained from [16, 27, 28] and will explain why sparsity is being promoted by the ARD algorithm.

Perhaps surprisingly, we shall show that the variational algorithm discussed above still preserves the principles of ARD, despite the fact that it is not based on maximizing the evidence, but rather on maximizing the ELBO (lower bound for the evidence). We call this framework “Variational ARD” (**VARD**).

2.3. Alternating-Minimization Algorithm for Variational ARD. As mentioned in Sec. 2.2, the true posterior distribution $p(\mathbf{x}|\mathbf{y}, \gamma)$ corresponding to the Poisson model in (1.1) and the prior in (2.1) is intractable, so we cannot solve the E-step in (2.7) exactly, and an approximation to the posterior should be used instead. Perhaps the most common approach is to use a free-form factorized posterior distribution (mean field variational Bayes) [37], however this is also intractable here due to the lack of conditional conjugacy. Instead, we restrict the posterior to a parametric multivariate Gaussian form. In addition, in order to reduce the computational complexity, we restrict ourselves to factorized distributions, i.e.,

$$(2.9) \quad \mathcal{D} \triangleq \{h(\mathbf{x}) \mid h(\mathbf{x}) = \prod_{i=1}^p \mathcal{N}(x_i|m_i, v_i), \quad \text{for some } m_i, v_i, \quad i = 1, \dots, p\},$$

where $\mathcal{N}(x_i|m_i, v_i)$ denotes a normal univariate distribution with mean m_i and variance v_i . We propose to modify the EM algorithm in (2.7)–(2.8) by adding the constraint $q \in \mathcal{D}$ during

the E-step in (2.7). The resulting algorithm is defined by the following two steps

(2.10)

$$\mathbf{B}: (\mathbf{m}^{(t+1)}, \mathbf{v}^{(t+1)}) = \arg \min_{\mathbf{m} \geq 0, \mathbf{v} \geq 0} \mathbb{E}_{q(\mathbf{x}; \mathbf{m}, \mathbf{v})} [-\log p(\mathbf{y}|\mathbf{x})] + D_{KL}[q(\mathbf{x}; \mathbf{m}, \mathbf{v}) || p(\mathbf{x}|\gamma^{(t)})],$$

(2.11)

$$\mathbf{F}: \gamma^{(t+1)} = \arg \min_{\gamma} D_{KL}[q(\mathbf{x}; \mathbf{m}^{(t+1)}, \mathbf{v}^{(t+1)}) || p(\mathbf{x}|\gamma)],$$

which are repeated until convergence. Note that we have rewritten the FVE of (2.5) as $\mathcal{F} = \mathbb{E}_{q(\mathbf{x}; \mathbf{m}, \mathbf{v})} [-\log p(\mathbf{y}|\mathbf{x})] + D_{KL}[q(\mathbf{x}; \mathbf{m}, \mathbf{v}) || p(\mathbf{x}|\gamma)]$. The first term in (2.10) does not depend on γ and was therefore omitted in (2.11). In the backward (B) step in (2.10), we update the approximation to the posterior q , which is reduced to only searching for the variational parameters (\mathbf{m}, \mathbf{v}) . In the forward (F) step in (2.11), we update the generative (forward) model by finding γ . Note that due to the constraint $q \in \mathcal{D}$ in (2.9), this is no longer an EM algorithm, and therefore we distinguish between the “E-step” and the “B-step,” and similarly between the “M-step” and the “F-step.”

Note that the posterior mean \mathbf{m} is the main object of interest and provides the estimates for the linear attenuation coefficients, which must be nonnegative. Accordingly, a non-negativity constraint on \mathbf{m} was added in (2.10). The function in (2.10) is only defined for positive \mathbf{v} values and using the extended-value function we assume $\mathcal{F} = \infty$ for any $v_j \leq 0$.

The AM algorithm formed by repeating the updates in (2.10)–(2.11) will reduce the FVE at each iteration (and increase the ELBO). Importantly, as shown below, the AM algorithm leads to near-sparse solutions, i.e., many of the marginal posterior distributions will be highly concentrated around zero in the transform space defined by (2.2). The roles of \mathbf{v} and γ in promoting sparse solutions will be made clear in Sec. 2.4.

For the Poisson model in (1.1), the prior in (2.1) and the approximate posterior proposed in (2.9), the objective function in (2.10) can be written as (dropping iteration number)

$$(2.12) \quad \mathcal{F} = \underbrace{\mathcal{F}_1(\mathbf{m}, \mathbf{v})}_{=\mathbb{E}_{q(\mathbf{x}; \mathbf{m}, \mathbf{v})} [-\log p(\mathbf{y}|\mathbf{x})]} + \underbrace{\mathcal{F}_2(\mathbf{m}, \gamma) + \mathcal{F}_3(\mathbf{v}, \gamma)}_{=D_{KL}[q(\mathbf{x}; \mathbf{m}, \mathbf{v}) || p(\mathbf{x}|\gamma)],}$$

$$(2.13) \quad \mathcal{F}_1(\mathbf{m}, \mathbf{v}) = \sum_{i=1}^n \left[y_i \sum_{j=1}^p \phi_{ij} m_j + \eta_i \exp\left(-\sum_{j=1}^p \phi_{ij} m_j + \sum_{j=1}^p \phi_{ij}^2 v_j / 2\right) \right],$$

$$(2.14) \quad \mathcal{F}_2(\mathbf{m}, \gamma) = \frac{1}{2} \sum_{k=1}^p \gamma_k^{-1} \left(\sum_{j=1}^p \psi_{kj} m_j \right)^2,$$

$$(2.15) \quad \mathcal{F}_3(\mathbf{v}, \gamma) = \frac{1}{2} \sum_{k=1}^p \sum_{j=1}^p \gamma_k^{-1} \psi_{kj}^2 v_j - \frac{1}{2} \sum_{j=1}^p \log v_j + \frac{1}{2} \sum_{k=1}^p \log \gamma_k,$$

where $\phi_{ij} = (\Phi)_{ij}$ are the entries of the system matrix defined after (1.1), and $\psi_{ij} = (\Psi)_{ij}$ are the entries of the Ψ matrix defined in (2.1). In deriving the last term in (2.15), we assumed that Ψ is square and non-singular⁵. We provide a generalization in Sec. 6 which removes this

⁵In deriving (2.15) we used $\log |(\Psi^T \Gamma^{-1} \Psi)^{-1}| = -\log |\Psi^T \Gamma^{-1} \Psi| = \log |\Gamma| + C = \sum_k \log \gamma_k + C$, where C is a finite constant and can be excluded. Note that $(\Psi^T \Gamma^{-1} \Psi)$ has to be non-singular in order for this expression to be well-defined. The second equality with $C < \infty$ only holds when Ψ is square and non-singular.

restriction and allows any Ψ .

Recalling the discussion in Sec. 1.1 about the sparse structure of ϕ_i , we note that the objective function in (2.12)–(2.15) has a computational complexity of $O(np^{1/D})$ ⁶ which is feasible to compute in order to assess convergence in practice. In contrast, the original ARD [15] or other extensions [28, 30] require $O(p^3)$ operations to evaluate the objective which is prohibitive for large scale problems where p is very large, so the objective can only be approximated, e.g., via the stochastic estimator of Papandreou and Yuille [33].

The solution to the F-step in (2.11) is obtained by solving $\nabla_{\gamma} D_{KL}[q||p] = \mathbf{0}$ which yields

$$(2.16) \quad \gamma^{(t+1)} = [\Psi \mathbf{m}^{(t+1)}] \odot [\Psi \mathbf{m}^{(t+1)}] + [\Psi \odot \Psi] \mathbf{v}^{(t+1)},$$

where \odot denotes elementwise product. It is straightforward to verify that (2.16) is indeed a minimizing point of (2.11) (for finite γ). For fixed γ , the FVE objective function is jointly convex with respect to (\mathbf{m}, \mathbf{v}) , which implies that any local minimum of the objective in (2.10) is also a global minimum. Note that the FVE function is not jointly convex with respect to $(\mathbf{m}, \mathbf{v}, \gamma)$, so local minima are possible. In a similar way, the original ARD-EM algorithm [15] is also only guaranteed to find a *local* maximum of the evidence [40].

2.4. Discussion. The B-step in (2.10) is well understood from a Bayesian variational inference perspective, and it is equivalent to minimizing the KL divergence between the proposed posterior $q(\mathbf{x})$ and the true posterior $p(\mathbf{x}|\mathbf{y}, \gamma)$, as described in Fig. 3(a).

However, the F-step in (2.11) is somewhat puzzling, as it is unclear why it promotes sparse solutions, especially since using a Gaussian prior for MAP estimation does not lead to sparse solutions. To explain how sparsity is promoted, consider the \mathbf{s} -space defined in (2.2), where the image has a sparse representation. Note that the KL divergence in (2.11) is invariant under the transformation in (2.2). The prior and posterior distributions become

$$(2.17) \quad p(\mathbf{s}|\gamma) = \mathcal{N}(\mathbf{s}|\mathbf{0}, \gamma), \quad q(\mathbf{s}) = \mathcal{N}(\mathbf{s}|\boldsymbol{\mu}, \boldsymbol{\Sigma}), \quad \boldsymbol{\mu} = \Psi \mathbf{m}, \quad \boldsymbol{\Sigma} = \Psi \text{Diag}(\mathbf{v}) \Psi^T,$$

where Ψ is assumed to be invertible. Next, we make use of the following representation for a factorized Student's-t distribution [27]

$$(2.18) \quad t(\mathbf{s}|\nu) \triangleq \prod_{j=1}^p t(s_j|\nu) = \prod_{j=1}^p \max_{\gamma_j \geq 0} \mathcal{N}(s_j|0, \gamma_j) \varphi(\gamma_j|\nu),$$

where $t(s_j|\nu) = \Gamma[(\nu+1)/2](1+s_j^2/\nu)^{-(\nu+1)/2}/[\sqrt{\nu\pi} \Gamma(\nu/2)]$ is a univariate Student's-t distribution with $\Gamma(\cdot)$ denoting the Gamma function, and $\varphi(\gamma_j|\nu) = \gamma_j^{-\nu/2} \exp(-\nu/2\gamma_j)$. To obtain (2.18), one exploits Fenchel duality [41] $\log t(\sqrt{s_j}|\nu) = \max_{\lambda_j \leq 0} (\lambda_j s_j - f^*(\lambda_j))$ where $f^*(\lambda_j)$ is the Fenchel dual of $\log t(\sqrt{s_j}|\nu)$, and then uses the monotonically increasing transformation $\lambda_j = -1/2\gamma_j$ to obtain $\log t(\sqrt{s_j}|\nu) = \max_{\gamma_j \geq 0} (-s_j/2\gamma_j + f^*(-1/2\gamma_j))$. By exponentiating both sides of the dual representation for $\log t(s_i|\nu)$ one obtains (2.18) [27]. For $\nu \rightarrow 0$, $t(s_i|\nu) \propto 1/|s_i|$ with sharp spikes at $s_i = 0$. Note in (2.18) that $\nu \rightarrow 0$ implies $\varphi \rightarrow 1$, and the multivariate t-distribution $t(\mathbf{s}|\nu)$ becomes an envelope for the ARD prior $p(\mathbf{s}|\gamma)$ in (2.17).

⁶For convenience we restate the definitions here: n is the number of measurements, p is the total number of pixels/voxels, and $D = 2, 3$ for 2D/3D images, respectively.

This is illustrated in Fig. 3(b) using a 2D example with $\mathbf{s} \in \mathbb{R}^2$, which depicts the proposed posterior $q(\mathbf{s})$ and the prior $p(\mathbf{s}|\gamma)$, with the latter “trapped” inside the level sets of $t(\mathbf{s}|\nu)$ due to (2.18). One can see in Fig. 3(b) how selecting the widths γ of the Gaussian prior $p(\mathbf{s})$ to minimize the KL divergence between $q(\mathbf{s})$ and $p(\mathbf{s})$ promotes the shrinking of γ_k for some k (minimizing KL divergence between the two distributions is illustrated as maximizing the overlap between the level-set ellipsoids corresponding to the majority of the probability mass of these distributions). In the example of Fig. 3, the posterior has a much lower mean and variance along s_2 than along s_1 , and the F-step will further shrink γ_2 . The corresponding parameter s_2 will be shrunk in the sense that the marginal prior $p(s_2|\gamma_2) = \mathcal{N}(s_2|0, \gamma_2)$ will be more concentrated around zero, and so will the true posterior $p(s_2|\mathbf{y})$ (see (1.3)). In the following B-step, the *approximate* posterior q will also concentrate further around zero due to the zero-forcing property of the KL divergence [42] (recall that in the B-step we minimize KL-divergence between q and the true posterior). Note that repeating the B and F steps is required to obtain sparse solutions.

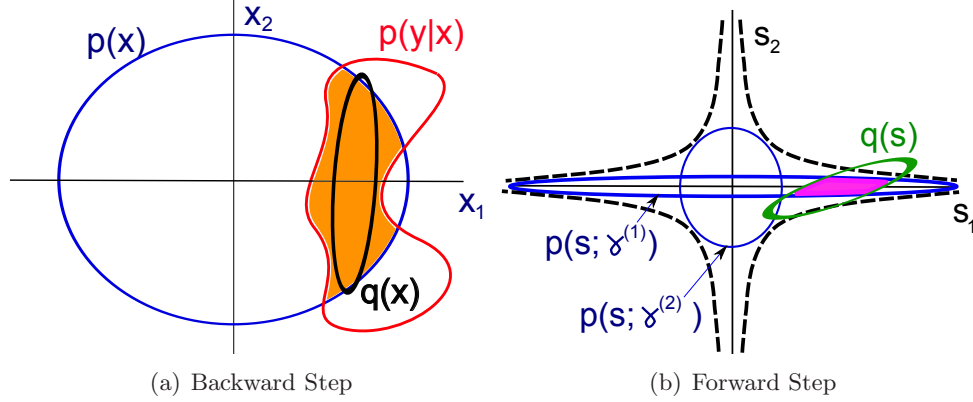


Figure 3. Description of the AM algorithm proposed in (2.10)–(2.11). (a) Backward step: level sets for the likelihood $p(\mathbf{y}|\mathbf{x})$ (red curve), prior $p(\mathbf{x})$ (blue curve) and the proposed posterior $q(\mathbf{x})$ (black curve); true posterior $p(\mathbf{x}|\mathbf{y})$ is concentrated in the orange region; proposed posterior $q(\mathbf{x})$ is chosen to minimize the KL divergence with $p(\mathbf{x}|\mathbf{y})$. (b) Forward step: a level set for the student’s t distribution (dashed black curves) forming an envelope for the level set of the zero mean Gaussian prior $p(\mathbf{s})$ (blue curves); $p(\mathbf{s}|\gamma^{(1)})$ and $p(\mathbf{s}|\gamma^{(2)})$ correspond to two different choices for γ where $p(\mathbf{s}|\gamma^{(2)})$ assigns equal variance along s_1 and s_2 and $p(\mathbf{s}|\gamma^{(1)})$ assigns small variance along s_2 and large variance along s_1 . Choosing $p(\mathbf{s}|\gamma^{(1)})$ maximizes the overlap between the prior $p(\mathbf{s})$ and the proposed posterior $q(\mathbf{s})$ (leading to lower KL divergence). s_2 is shrunk in the sense that its exact and proposed posterior distributions are further concentrated around smaller values due to the concentration of the prior about zero. Note that Fig. (a) is in the \mathbf{x} domain whereas Fig. (b) is in the \mathbf{s} domain.

The description of the mechanism responsible for promoting sparsity was inspired by the work of Wipf et al.[27], which uses similar arguments. However, that work relies entirely on the evidence function $p(\mathbf{y}|\gamma)$ in (2.4) since a Gaussian noise model was considered. In contrast, our proposed approach is based on the free variational energy (FVE) function and applies more generally, even to cases where the expression for the evidence is unknown and an algorithm that increases the evidence is unavailable. This is important for non-Gaussian noise models, for which the evidence function is intractable, so we are forced to work with the FVE function instead. Another important difference between our work and that of Wipf et al.[27]

is that we show how sparsity is promoted *during the iterations of the algorithm*, whereas in [27] it has been shown that a solution to (1.7) is sparse. Note also that [27] considers the original ARD prior in (1.6), whereas we consider the more general case in (2.1).

3. Parallel AM Algorithm. Here we present a modified AM algorithm, which uses an alternative B-step. The rationale behind the modified algorithm lies in *decoupling* the variational parameters m_j, v_j of each pixel/voxel. This reduces the algorithm into simple 1D optimization tasks that can be computed in parallel, leading to high scalability. First, note that (2.13) and (2.14) are not additively separable with respect to the individual parameters m_j, v_j . We shall therefore introduce a surrogate function, making use of the following lemma.

Lemma 3.1 (The convex decomposition lemma [13]). *Let $f(\mathbf{x})$ be a convex function with $\mathbf{x} \in \mathbb{R}^p$ and let $\hat{\mathbf{x}} \in \mathbb{R}^p$ be a reference point. Let $r_j \geq 0$ such that $\sum_{j=1}^p r_j = 1$. Then*

$$(3.1) \quad f(\mathbf{x}) \leq \sum_{j=1}^p r_j f(\hat{\mathbf{x}} + \frac{x_j - \hat{x}_j}{r_j} \mathbf{e}_j),$$

where \mathbf{e}_j denotes a column vector with all zeros except 1 at the j th entry. **Equality** in (3.1) is obtained if $x_j = \hat{x}_j$ for all j or if f is affine.

Proof. The proof readily follows from Jensen's inequality [41]

$$(3.2) \quad f(\mathbf{x}) = f(\hat{\mathbf{x}} + \sum_{j=1}^p (x_j - \hat{x}_j) \mathbf{e}_j) = f(\hat{\mathbf{x}} + \sum_{j=1}^p r_j \frac{x_j - \hat{x}_j}{r_j} \mathbf{e}_j) \leq \sum_{j=1}^p r_j f(\hat{\mathbf{x}} + \frac{x_j - \hat{x}_j}{r_j} \mathbf{e}_j),$$

where we added and subtracted a reference point $\hat{\mathbf{x}}$ and then applied Jensen's inequality to $g(\sum_j r_j [\boldsymbol{\nu}_j / r_j])$ where $\boldsymbol{\nu}_j \triangleq (x_j - \hat{x}_j) \mathbf{e}_j$ and $g(\mathbf{x}) \triangleq f(\hat{\mathbf{x}} + \mathbf{x})$. Equality is obtained iff $\boldsymbol{\nu}_j / r_j$ is independent of j or f is affine. The former is obtained if $x_j = \hat{x}_j, \forall j$. ■

We will also use a minor (but useful) extension of lemma 3.1 by O'Sullivan and Benac [13], where we introduce a dummy variable denoted x_0 with a corresponding weight $r_0 \geq 0$, such that $f(x_0, \mathbf{x}) = f(\mathbf{x})$. Accordingly, \mathbf{x} and \mathbf{r} in (3.1) are replaced by (x_0, \mathbf{x}) and (r_0, \mathbf{r}) , respectively (\mathbf{r} is the collection of all r_j 's), and the summation in (3.1) over j from 1 to p is replaced by a summation from 0 to p . This modification leads to the requirement that $\sum_{j=1}^p r_j + r_0 = 1$ which allows the freedom to choose $\sum_{j=1}^p r_j \leq 1$ instead of strict equality as in Lemma 3.1 (which of course requires that $r_0 = 1 - \sum_{j=1}^p r_j$). We choose to fix the dummy variable $x_0 = \hat{x}_0$ so that the term corresponding to $j = 0$ on the right hand side of (3.1) is just a constant and will not be involved in the following optimization.

Before we proceed, we make several definitions to simplify the following derivations,

$$(3.3) \quad p_i^{(t)} \triangleq \sum_j \phi_{ij} m_j^{(t)}, \quad \tilde{p}_i^{(t)} \triangleq \sum_j \phi_{ij}^2 v_j^{(t)},$$

$$(3.4) \quad \mu_i^{(t)} \triangleq \mathbb{E}_{q(\mathbf{x}; \mathbf{m}^{(t)}, \mathbf{v}^{(t)})} [\eta_i \exp(-\boldsymbol{\phi}_i^T \mathbf{x})] = \eta_i \exp(\tilde{p}_i^{(t)} / 2) \exp(-p_i^{(t)}),$$

$$(3.5) \quad b_j^y \triangleq \sum_i \phi_{ij} y_i, \quad b_j^{(t)} \triangleq \sum_i \phi_{ij} \mu_i^{(t)}, \quad \tilde{b}_j^{(t)} \triangleq \sum_i \phi_{ij}^2 \mu_i^{(t)} / 2,$$

where y_i, ϕ_i and η_i are defined in (1.1). To distinguish between quantities associated with the mean and variance, we denote any quantities associated with the variance using a tilde. In

(3.3), $p_i^{(t)}$ is the projection of the posterior mean $\mathbf{m}^{(t)}$ along the i th line, which we shall call “mean-type” projection; $\tilde{p}_i^{(t)}$ is a projection of the posterior variance $\mathbf{v}^{(t)}$ along the i th line using an elementwise squared system matrix, which we term a “variance-type” projection. In (3.4), $\mu_i^{(t)}$ is the predicted Poisson rate for the i th measurement, based on the expectation with respect to posterior distribution q computed at iteration t . In (3.5), b_j^y is the back-projection of measurements \mathbf{y} to the j th pixel; $b_j^{(t)}$ and $\tilde{b}_j^{(t)}$ are the back-projections of the predicted Poisson rate in (3.4) to the j th pixel, using the backward (adjoint) operator and the squared backward operator, respectively, which we call “mean-type” and “variance-type” backprojections.

By noting that (2.13) can be written as $\mathcal{F}_1 = \sum_i \mathcal{F}_{1i}$ and that \mathcal{F}_{1i} is jointly convex with respect to (\mathbf{m}, \mathbf{v}) , we apply Lemma 3.1 to each \mathcal{F}_{1i} separately. Using (3.1) with its minor extension (mentioned after Lemma 3.1) and $\mathbf{x} = (\mathbf{m}, \mathbf{v})$, $\hat{\mathbf{x}} = (\mathbf{m}^{(t)}, \mathbf{v}^{(t)})$, and choosing a dummy variable m_0 which is set to zero, we obtain the bound $\mathcal{F}_1 \leq \mathcal{S}_1^{(t)}(\mathbf{m}, \mathbf{v})$ at iteration t with $\mathcal{S}_1^{(t)}$ defined as

$$(3.6) \quad \mathcal{S}_1^{(t)}(\mathbf{m}, \mathbf{v}) = \sum_i \sum_j y_i \phi_{ij} m_j + \mu_i^{(t)} r_{ij} \exp \left[-\frac{\phi_{ij}}{r_{ij}} (m_j - m_j^{(t)}) \right] + \mu_i^{(t)} \tilde{r}_{ij} \exp \left[\frac{\phi_{ij}^2}{\tilde{r}_{ij}} (v_j - v_j^{(t)})/2 \right],$$

where we made use of (3.3)–(3.5) and r_{ij} corresponds to m_j while \tilde{r}_{ij} corresponds to v_j (for the i th measurement). Note that the summation over j in (3.6) includes the dummy variable m_0 with $m_0 = m_0^{(t)} = 0$ for any t . Also note that (3.6) is separable with respect to the components m_j and v_j . According to the extended Lemma 3.1 we are free to choose r_{ij}, \tilde{r}_{ij} as long as

$$(3.7) \quad \sum_j r_{ij} + \sum_j \tilde{r}_{ij} \leq 1,$$

which must be satisfied for each i . We make the following choice for r_{ij} and \tilde{r}_{ij}

$$(3.8) \quad r_{ij} = \frac{\phi_{ij}}{Z_1}, \quad \tilde{r}_{ij} = \frac{\phi_{ij}^2}{2Z_1}, \quad Z_1 = \max_i \sum_j \phi_{ij} + \phi_{ij}^2/2,$$

which can be easily verified to satisfy the condition in (3.7). Substituting (3.8) into (3.6), and using the definitions in Eq. (3.5), we obtain

$$(3.9) \quad \mathcal{S}_1^{(t)}(\mathbf{m}, \mathbf{v}) = \sum_j b_j^y m_j + \frac{b_j^{(t)}}{Z_1} \exp \left[-Z_1 (m_j - m_j^{(t)}) \right] + \frac{\tilde{b}_j^{(t)}}{Z_1} \exp \left[Z_1 (v_j - v_j^{(t)}) \right].$$

The particular choice in (3.8) is very useful since the sum over i in (3.6) needs to be done only once instead of repeating it for each update of m_j during the search for the minimum of \mathcal{S}_1 .

We derive a separable bound for (2.14), $\mathcal{F}_2 \leq \mathcal{S}_2^{(t)}(\mathbf{m}, \mathbf{v})$, by noting that x^2 is a convex function and using lemma 3.1 again to obtain

$$(3.10) \quad \mathcal{S}_2^{(t)}(\mathbf{m}, \mathbf{v}) = \frac{1}{2} \sum_k \frac{1}{\gamma_k^{(t)}} \sum_j s_{kj} \left[(d_k^{(t)})^2 + 2d_k^{(t)} \frac{\psi_{kj}}{s_{kj}} (m_j - m_j^{(t)}) + \frac{\psi_{kj}^2}{s_{kj}^2} (m_j - m_j^{(t)})^2 \right],$$

where $d_k^{(t)} \triangleq \sum_j \psi_{kj} m_j^{(t)}$ (also defined in (3.13)). Similarly to (3.7), we require $\sum_j s_{kj} \leq 1$ using the dummy variable extension. One choice for s_{kj} is

$$(3.11) \quad s_{kj} = \frac{|\psi_{kj}|}{Z_2}, \quad Z_2 = \max_k \sum_j |\psi_{kj}|.$$

Substituting (3.11) into (3.10), we obtain

$$(3.12) \quad \mathcal{S}_2^{(t)}(\mathbf{m}, \mathbf{v}) = h^{(t)} + \sum_j [f_j^{(t)} (m_j - m_j^{(t)}) + g_j^{(t)} (m_j - m_j^{(t)})^2],$$

where we introduced the following definitions

$$(3.13) \quad d_k^{(t)} \triangleq \sum_j \psi_{kj} m_j^{(t)}, \quad f_j^{(t)} \triangleq \sum_k \psi_{kj} d_k^{(t)} / \gamma_k^{(t)},$$

$$(3.14) \quad g_j^{(t)} \triangleq Z_2 \sum_k |\psi_{kj}| / 2\gamma_k^{(t)}, \quad h^{(t)} \triangleq \sum_k (d_k^{(t)})^2 / 2\gamma_k^{(t)},$$

$$(3.15) \quad \xi_j^{(t)} \triangleq \sum_k \psi_{kj}^2 / \gamma_k^{(t)}.$$

$d_k^{(t)}$ in (3.13) is the estimated k th element in the sparse representation for the object based on the posterior mean $\mathbf{m}^{(t)}$, e.g., if Ψ is chosen according to (2.3) (assuming sparsity in the pixel/voxel-difference space) then $d_k^{(t)} = m_k^{(t)} - \sum_{j: \{j,k\} \in \Theta} [m_j^{(t)} / N]$ where Θ defines the neighborhood system and N is the number of neighbors. In deriving (3.14) we made use of $\sum_j s_{kj} = 1$.

Combining (3.9), (3.12), and (2.15) (the latter is already separable), we obtain a surrogate function $\mathcal{S}^{(t)}$ that is a global majorizer of \mathcal{F} , i.e., $\mathcal{F}(\mathbf{m}, \boldsymbol{\gamma}) \leq \mathcal{S}^{(t)}(\mathbf{m}, \mathbf{v})$, given by

$$(3.16) \quad \mathcal{S}^{(t)}(\mathbf{m}, \mathbf{v}) = \sum_j \mathcal{S}_{m_j}^{(t)}(m_j) + \sum_j \mathcal{S}_{v_j}^{(t)}(v_j),$$

$$(3.17) \quad \mathcal{S}_{m_j}^{(t)}(m_j) = b_j^y m_j + \frac{b_j^{(t)}}{Z_1} \exp[-Z_1(m_j - m_j^{(t)})] + f_j^{(t)} (m_j - m_j^{(t)}) + g_j^{(t)} (m_j - m_j^{(t)})^2,$$

$$(3.18) \quad \mathcal{S}_{v_j}^{(t)}(v_j) = \frac{\tilde{b}_j^{(t)}}{Z_1} \exp[Z_1(v_j - v_j^{(t)})] + \xi_j^{(t)} v_j / 2 - \log v_j / 2.$$

Note that the surrogate function \mathcal{S} in (3.16) is separable with respect to both the components of \mathbf{m} and \mathbf{v} . This yields a modified B-step where \mathcal{S} is minimized by simultaneous (parallel) 1D optimization tasks, each with respect to a different component m_j or v_j . The iterations are defined as

$$(3.19) \quad m_j^{(t+1)} = \arg \min_{m_j \geq 0} \mathcal{S}_{m_j}^{(t)}(m_j),$$

$$(3.20) \quad v_j^{(t+1)} = \arg \min_{v_j \geq 0} \mathcal{S}_{v_j}^{(t)}(v_j).$$

Algorithm 1 : PAR-VARD (VARD with Separable Surrogates)

| | |
|---|---|
| 1: Initialize $\mathbf{m}^{(1)}, \mathbf{v}^{(1)}, \gamma^{(1)}$ | |
| 2: Compute \mathbf{b}^y in (3.5) | % backprojection of data % |
| 3: for t=1 to N do | % AM iterations % |
| 4: $\mathbf{p}^{(t)} \leftarrow \Phi \mathbf{m}^{(t)}$ | % “mean-type” projections % |
| 5: $\tilde{\mathbf{p}}^{(t)} \leftarrow (\Phi \odot \Phi) \mathbf{v}^{(t)}$ | % “variance-type” projections % |
| 6: $\boldsymbol{\mu}^{(t)} \leftarrow \boldsymbol{\eta} \odot \exp(\tilde{\mathbf{p}}^{(t)}/2) \odot \exp(-\mathbf{p}^{(t)})$ | |
| 7: Compute $\mathbf{g}^{(t)}, \boldsymbol{\xi}^{(t)}$ and $\mathbf{f}^{(t)}$ defined in (3.13)–(3.15) | |
| 8: $\mathbf{b}^{(t)} \leftarrow \Phi^T \boldsymbol{\mu}^{(t)}$ | % “mean-type” backprojections of $\boldsymbol{\mu}^{(t)}$ % |
| 9: $\tilde{\mathbf{b}}^{(t)} \leftarrow (\Phi \odot \Phi)^T \boldsymbol{\mu}^{(t)}$ | % “variance-type” backprojections of $\boldsymbol{\mu}^{(t)}$ % |
| 10: for j=1 to p do | % executed in parallel % |
| 11: $m_j^{(t+1)} \leftarrow \arg \min_{m_j \geq 0} \mathcal{S}_{m_j}^{(t)}(m_j; m_j^{(t)}, b_j^{(t)}, f_j^{(t)}, g_j^{(t)})$ | % see (3.17) % |
| 12: $v_j^{(t+1)} \leftarrow \arg \min_{v_j \geq 0} \mathcal{S}_{v_j}^{(t)}(v_j; v_j^{(t)}, \tilde{b}_j^{(t)}, \xi_j^{(t)})$ | % see (3.18) % |
| 13: $\gamma^{(t+1)} \leftarrow [\Psi \mathbf{m}^{(t)}] \odot [\Psi \mathbf{m}^{(t)}] + [\Psi \odot \Psi] \mathbf{v}^{(t)}$ | % see (2.16) % |

Combining the modified B-step in (3.19)–(3.20) with the F-step in (2.16), we obtain the parallel VARD algorithm described in Algorithm 1. Each iteration requires a “mean-type” forward-projection $\mathbf{p}^{(t)}$ and a “variance type” forward-projection $\tilde{\mathbf{p}}^{(t)}$ (lines 4-5 in Algorithm 1) which can be computed in parallel. It also requires a “mean-type” backprojection $\mathbf{b}^{(t)}$ and a “variance-type” backprojection $\tilde{\mathbf{b}}^{(t)}$ (lines 8-9 in Algorithm 1), which can also be computed in parallel. Each iteration also requires solving two types of 1D optimization problems, one for the mean components m_j and one for the variance components v_j (lines 11-12 in Algorithm 1). The 1D optimization tasks for all pixels/voxels can be done in parallel and the scalar updates for mean and variance can also be done in parallel. Note that despite the latter parallelization, $\mathbf{b}^{(t)}$ and $\tilde{\mathbf{b}}^{(t)}$ provide information about both the mean and variance at the previous iteration which is shared during the mean and variance updates in (3.19)–(3.20) (lines 11-12 in Algorithm 1). Also note that these subiterations for (\mathbf{m}, \mathbf{v}) are themselves iterations of another alternating minimization algorithm (lines 4-12 of Algorithm 1). Here we chose to use only one subiteration, but any number can be used instead.

A considerable advantage of this algorithm is that it allows fast parallel computing on multi-core computers, or on dedicated hardware such as field-programmable gate arrays (FPGA). Another advantage is that the solution to the constrained problem in (3.19) can be obtained by first solving the unconstrained problem using simple gradient-based methods; if $m_j < 0$ then the solution to the constrained problem must be $m_j = 0$, otherwise the solution is the same. This simple thresholding for m_j leads to the optimal solution only in one-dimensional minimization problems, and the same procedure for the original B-step in (2.10) can lead to sub-optimal solutions or even lack of convergence.

The computational complexity of a forward/backprojection is $O(np^{1/D})$, where n is the number of measurements, p is the total number of pixels/voxels and $D = 2, 3$ for 2D/3D images, respectively (recall the discussion about the sparsity of Φ in Sec. 1.1). The computational complexity of the operations $\Psi \mathbf{m}$ and $(\Psi \odot \Psi) \mathbf{v}$ is $O(p(N+1))$, where N is the number

of neighboring pixels/voxels defined by the prior (recall the discussion on the sparsity of Ψ in Sec. 2.1). Typically, $N \approx 3^D - 1$, $n \geq p$, and $p^{1/D} \gg N$, so the forward/backprojection cost dominates the cost of $\Psi \mathbf{m}$ and $(\Psi \odot \Psi) \mathbf{v}$. The details regarding the implementations of the 1D optimization tasks (lines 11-12 in Algorithm 1) are given in Appendix B.

4. Optimality Conditions and Convergence Analysis. First we provide the necessary optimality conditions for the minimization problems presented in Secs. 2–3 as well as a few definitions in order to simplify the following analysis. We then present a detailed study of the convergence properties of the parallel AM algorithm derived in Sec. 3 (Algorithm 1). We shall refer to this algorithm as “PAR-VARD” (parallel VARD).

4.1. Necessary Optimality Conditions. Proposition 4.1.

1. The necessary Karush-Kuhn-Tucker (KKT) optimality conditions [41] for (\mathbf{m}, \mathbf{v}) to be a solution to (2.10) are given by

$$(4.1) \quad b_j^y + \sum_k \psi_{kj} d_k / \gamma_k^{(t)} - b_j \begin{cases} = 0 & \forall j \text{ s.t. } m_j > 0 \\ \geq 0 & \forall j \text{ s.t. } m_j = 0 \end{cases},$$

$$(4.2) \quad \tilde{b} = \xi_j^{(t)} - 1/v_j, \quad \forall j,$$

where b_j^y, b_j, \tilde{b}_j are defined by (3.5) with $m_j^{(t+1)}$ and $v_j^{(t+1)}$ replaced by m_j and v_j , respectively; d_k is defined by $d_k^{(t)}$ in (3.13) with $m_j^{(t)}$ replaced by m_j ; $\xi_j^{(t)}$ is defined in (3.15).

2. The necessary KKT condition for γ to be a solution to (2.11) is given by

$$(4.3) \quad \gamma = (\Psi \mathbf{m}^{(t+1)}) \odot (\Psi \mathbf{m}^{(t+1)}) + (\Psi \odot \Psi) \mathbf{v}^{(t+1)}.$$

3. The necessary KKT conditions for $(\mathbf{m}, \mathbf{v}, \gamma)$ to be a minimizer of the FVE objective in (2.12)–(2.15) subject to $\mathbf{m} \succeq 0$ are given by

$$(4.4) \quad b_j^y + f_j - b_j \begin{cases} = 0 & \forall j \text{ s.t. } m_j > 0 \\ \geq 0 & \forall j \text{ s.t. } m_j = 0 \end{cases},$$

$$(4.5) \quad \tilde{b} = \xi_j - 1/v_j, \quad \forall j,$$

$$(4.6) \quad \gamma = (\Psi \mathbf{m}) \odot (\Psi \mathbf{m}) + (\Psi \odot \Psi) \mathbf{v},$$

where ξ_j is defined as $\xi_j^{(t)}$ in (3.15) with $\gamma_j^{(t)}$ replaced by γ_j ; f_j is defined as $f_j^{(t)}$ in (3.13) with $d_k^{(t)}$ and $\gamma_k^{(t)}$ replaced by d_k and γ_k , respectively (see also definitions after (4.2)).

Proof. Follows directly from the KKT conditions $\nabla_{m_j} \mathcal{F} \geq 0 \forall j \text{ s.t. } m_j = 0$, $\nabla_{m_j} \mathcal{F} = 0 \forall j \text{ s.t. } m_j > 0$, $\nabla_{\mathbf{v}} \mathcal{F} = \mathbf{0}$, $\nabla_{\gamma} \mathcal{F} = \mathbf{0}$, and by plugging in (2.12)–(2.15). For (1) substitute $\gamma = \gamma^{(t)}$ and use only KKT conditions for (\mathbf{m}, \mathbf{v}) ; For (2) substitute $(\mathbf{m}, \mathbf{v}) = (\mathbf{m}^{(t+1)}, \mathbf{v}^{(t+1)})$ and use KKT condition for γ . ■

Remark 1: note that $v_j > 0$ due to the $-\log v_j$ term in (2.15). **Remark 2:** note that (4.3) is the F-step update in (2.16) so it is satisfied at the end of each iteration of PAR-VARD.

Proposition 4.2.

1. The necessary KKT conditions for m_j to be a solution to (3.19) are

$$(4.7) \quad b_j^y - b_j^{(t)} \exp[-Z_1(m_j - m_j^{(t)})] + f_j^{(t)} + 2g_j^{(t)}(m_j - m_j^{(t)}) \begin{cases} = 0 & \forall j \text{ s.t. } m_j > 0 \\ \geq 0 & \forall j \text{ s.t. } m_j = 0 \end{cases},$$

where b_j^y , $b_j^{(t)}$, $f_j^{(t)}$, $g_j^{(t)}$ and Z_1 are defined in (3.5), (3.13), (3.14) and (3.8), respectively.

2. The necessary KKT condition for v_j to be a solution to (3.20) is

$$(4.8) \quad \tilde{b}_j^{(t)} \exp[Z_1(v_j - v_j^{(t)})] + \xi_j^{(t)}/2 - 1/2v_j = 0,$$

where $\tilde{b}_j^{(t)}$, $\xi_j^{(t)}$ are defined in (3.5) and (3.15).

Proof. Conditions (4.7)–(4.8) follow directly from the KKT conditions: $\nabla_{m_j} \mathcal{S}_{m_j} \geq 0 \forall j$ s.t. $m_j = 0$, $\nabla_{m_j} \mathcal{S}_{m_j} = 0 \forall j$ s.t. $m_j > 0$, $\nabla_{v_j} \mathcal{S}_{v_j} = 0 \forall j$, and by plugging in (3.17)–(3.18). ■

Remark: Note that $v_j > 0$ due to the $-\log v_j$ term in (3.18).

4.2. Definitions. To simplify the convergence study in Sec. 4.3, we introduce a few definitions concerning the solutions to (3.19)–(3.20).

Definition 4.3. Let $\bar{m}_j^{(t)}$ be a solution to (3.19) but without the non-negativity constraint. The corresponding solution to (3.19) with the non-negativity constraint can be expressed as

$$(4.9) \quad m_j^{(t+1)} = \left[\bar{m}_j^{(t)} - \frac{1}{Z_1} \log \left(\frac{\beta_j^{(t)}}{b_j^{(t)}} \right) \right]_+,$$

where $[x]_+ = \max[0, x]$ and $\beta_j^{(t)}$ is defined as

$$(4.10) \quad \beta_j^{(t)} \triangleq b_j^y + f_j^{(t)} + 2g_j^{(t)}(\bar{m}_j^{(t)} - m_j^{(t)}).$$

Remark: note from (3.5) that $b_j^{(t)} \geq 0$, and from (4.7) and (4.10) it follows that $\beta_j^{(t)} \geq 0$, so the logarithm in (4.9) is well defined.

Definition 4.4. The solution to (3.19) can alternatively be expressed as

$$(4.11) \quad m_j^{(t+1)} = m_j^{(t)} - \frac{1}{Z_j^{(t)}} \log \left(\frac{\beta_j^{(t)}}{b_j^{(t)}} \right),$$

where we define

$$(4.12) \quad Z_j^{(t)} \triangleq \begin{cases} Z_1 & \text{for } \bar{m}_j^{(t)} \geq 0 \\ \frac{1}{m_j^{(t)}} \log \left(\frac{\beta_j^{(t)}}{b_j^{(t)}} \right) & \text{for } \bar{m}_j^{(t)} < 0 \text{ and } m_j^{(t)} > 0 \\ \infty & \text{for } \bar{m}_j^{(t)} < 0 \text{ and } m_j^{(t)} = 0. \end{cases}$$

Remark 1: introducing $Z_j^{(t)}$ in (4.11) has the same effect as thresholding in (4.9). **Remark 2:** note that since $m^{(t+1)} \geq \bar{m}^{(t)}$ we have $Z_j^{(t)} \geq Z_1$. We also introduce a similar definition for the variance.

Definition 4.5. The solution to (3.20) can be expressed as

$$(4.13) \quad v_j^{(t+1)} = v_j^{(t)} + \frac{1}{Z_1} \log \left(\frac{\tilde{\beta}_j^{(t)}}{b_j^{(t)}} \right),$$

where $\tilde{\beta}_j^{(t)}$ is defined as

$$(4.14) \quad \tilde{\beta}_j^{(t)} \triangleq \frac{1}{2v_j^{(t+1)}} - \xi_j^{(t+1)}/2.$$

Remark: since $\tilde{b}_j^{(t)} \geq 0$ and $\xi_j^{(t)} \geq 0$, it follows from (4.8) that $v_j^{(t+1)} > 0$.

Definition 4.6. The I-divergence between two vectors $\mathbf{p}, \mathbf{q} \in \mathbb{R}_+^M$ is defined as

$$(4.15) \quad I(\mathbf{p}||\mathbf{q}) = \sum_{j=1}^M p_j \log \left(\frac{p_j}{q_j} \right) - p_j + q_j.$$

We define $\log(\frac{0}{0}) = 0$. **Remark:** the I-divergence is nonnegative.

Definition 4.7. The Itakura-Saito (IS) divergence between two vectors $\mathbf{p}, \mathbf{q} \in \mathbb{R}_+^M$ is defined as

$$(4.16) \quad D_{IS}(\mathbf{p}||\mathbf{q}) = \sum_k [\log(p_k/q_k) + q_k/p_k - 1].$$

We define $\log(\frac{0}{0}) = 0$, and $\frac{0}{0} = 1$. **Remark:** D_{IS} is nonnegative.

4.3. Convergence Analysis. For simplicity we shall denote $\mathcal{F}^{(t)} \triangleq \mathcal{F}(\mathbf{m}^{(t)}, \mathbf{v}^{(t)}, \boldsymbol{\gamma}^{(t)})$.

Corollary 4.8. A fixed point of the iterates of PAR-VARD satisfies the necessary KKT conditions in (4.4)–(4.6) and also (4.1)–(4.3).

Proof. Substituting a fixed point $m_j = m_j^{(t)}$ and $\gamma_k = \gamma_k^{(t)}$ into (4.7) and (4.1) reduces (4.7) to (4.4) and (4.1). Substituting a fixed point $v_j = v_j^{(t)}$ and $\gamma_k = \gamma_k^{(t)}$ into (4.8) and (4.2) reduces (4.8) to (4.5) and (4.2). Recall the definitions after (4.2) and (4.6). The condition in (4.6) is obtained by substituting $\mathbf{m}^{(t+1)} = \mathbf{m}$ and $\mathbf{v}^{(t+1)} = \mathbf{v}$ into (4.3). ■

The following theorem is due to Degirmenci and O’Sullivan.

Theorem 4.9. The objective function of (2.12)–(2.15) decreases monotonically during the iterations of PAR-VARD, and the decrease between subsequent iterations is lower bounded by

$$(4.17) \quad \mathcal{F}^{(t)} - \mathcal{F}^{(t+1)} \geq I\left(\frac{\boldsymbol{\beta}^{(t)}}{\mathbf{Z}^{(t)}} \parallel \frac{\mathbf{b}^{(t)}}{\mathbf{Z}^{(t)}}\right) + I\left(\frac{\tilde{\boldsymbol{\beta}}^{(t)}}{Z_1} \parallel \frac{\tilde{\mathbf{b}}^{(t)}}{Z_1}\right) + \sum_j g_j^{(t)} \left[\frac{1}{Z_j^{(t)}} \log \left(\frac{\beta_j^{(t)}}{b_j^{(t)}} \right) \right]^2 + \frac{1}{2} D_{IS}(\boldsymbol{\gamma}^{(t)} || \boldsymbol{\gamma}^{(t+1)}),$$

where $\boldsymbol{\beta}^{(t)}$, $\tilde{\boldsymbol{\beta}}^{(t)}$, and $\mathbf{Z}^{(t)}$ are vectors with components $\beta_j^{(t)}$, $\tilde{\beta}_j^{(t)}$, and $Z_j^{(t)}$, defined in (4.10), (4.14) and (4.12), respectively. $\mathbf{b}^{(t)}$, $\tilde{\mathbf{b}}^{(t)}$ are vectors with components $b_j^{(t)}$ and $\tilde{b}_j^{(t)}$ defined in (3.5),

and $g_j^{(t)}$ is defined in (3.14) ($g_j^{(t)} \geq 0$). Dividing two vectors should be interpreted as done elementwise.

Proof. The proof is very technical and therefore moved to Appendix A. ■
Several properties follow from Theorem 4.9 as stated in Theorem 4.10.

Theorem 4.10. Assume that $\mathcal{F}^{(1)}$ is finite. Then

- (a) The I-divergence $I(\frac{\beta^{(t)}}{\mathbf{Z}^{(t)}} \parallel \frac{\mathbf{b}^{(t)}}{\mathbf{Z}^{(t)}})$ converges to zero.
- (b) The I-divergence $I(\frac{\tilde{\beta}^{(t)}}{\mathbf{Z}_1} \parallel \frac{\tilde{\mathbf{b}}^{(t)}}{\mathbf{Z}_1})$ converges to zero.
- (c) The set of limit points of the mean iterates $\mathbf{m}^{(t)}$ is a connected set.
- (d) The set of limit points of the variance iterates $\mathbf{v}^{(t)}$ is a connected set.

Proof. (a)-(b): From theorem 4.9 we have

$$(4.18) \quad \begin{aligned} \mathcal{F}^{(1)} - \mathcal{F}^{(t)} &= \sum_{s=1}^{t-1} [\mathcal{F}^{(s)} - \mathcal{F}^{(s+1)}] \geq \sum_{s=1}^{t-1} I\left(\frac{\beta^{(s)}}{\mathbf{Z}^{(s)}} \parallel \frac{\mathbf{b}^{(s)}}{\mathbf{Z}^{(s)}}\right) + \sum_{s=1}^{t-1} I\left(\frac{\tilde{\beta}^{(s)}}{\mathbf{Z}_1} \parallel \frac{\tilde{\mathbf{b}}^{(s)}}{\mathbf{Z}_1}\right) \\ &+ \sum_{s=1}^{t-1} \sum_j g_j^{(s)} \left[\frac{1}{Z_j^{(s)}} \log \left(\frac{\beta_j^{(s)}}{b_j^{(s)}} \right) \right]^2 + \sum_{s=1}^{t-1} \frac{1}{2} D_{IS}(\gamma^{(s)} \parallel \gamma^{(s+1)}). \end{aligned}$$

As $t \rightarrow \infty$ all term on the right-hand side of (4.18) must go to zero since they are nonnegative and have a finite positive upper bound due to the fact that $\mathcal{F}^{(1)}$ is finite and due to the monotonically decreasing and positive sequence $\mathcal{F}^{(t)}$.

(c)-(d): Connectedness of the limit set of the iterates follows from convergence of the I-divergences to zero which implies by Pinsker's inequality [46][Lemma 12.6.1] that $\|\beta^{(t)} - \mathbf{b}^{(t)}\|_1 \rightarrow 0$ and $\|\tilde{\beta}^{(t)} - \tilde{\mathbf{b}}^{(t)}\|_1 \rightarrow 0$. This together with (4.11) and (4.13) implies that $m_j^{(t+1)} - m_j^{(t)} \rightarrow 0 \forall j$, $v_j^{(t+1)} - v_j^{(t)} \rightarrow 0 \forall j$. ■

We are now able to state the guarantees for global⁷ convergence of PAR-VARD.

Theorem 4.11 (Global Convergence Theorem). Let $\{\mathbf{z}^{(t)}\}_{t=0}^\infty = \{(\mathbf{m}^{(t)}, \mathbf{v}^{(t)}, \gamma^{(t)})\}_{t=0}^\infty$ be the sequence of iterates generated by Algorithm 1. Let the solution set Γ be the set of points $(\mathbf{m}, \mathbf{v}, \gamma)$ which satisfy the KKT conditions in (4.4)–(4.6). Assume there exists $\mathbf{z}^{(1)}$ such that $\mathcal{F}^{(1)} = \mathcal{F}(\mathbf{z}^{(1)})$ is finite. Then

- (a) The iterates are contained in a compact set.
- (b) For $\mathbf{z}^{(t)} \in \Gamma$, $\mathcal{F}^{(t)} \geq \mathcal{F}^{(t+1)}$.
- (c) For $\mathbf{z}^{(t)} \notin \Gamma$, $\mathcal{F}^{(t)} > \mathcal{F}^{(t+1)}$.
- (d) The point-to-set mapping defined by Algorithm 1 is closed.
- (e) All limit points of the iterates are in the solution set Γ and $\mathcal{F}^{(t)} \rightarrow \mathcal{F}(\mathbf{m}, \mathbf{v}, \gamma)$ as $t \rightarrow \infty$ for some $(\mathbf{m}, \mathbf{v}, \gamma) \in \Gamma$.

Proof. (a) By assumption, there is an initial guess $\mathbf{z}^{(1)}$ such that $\mathcal{F}^{(1)} = \mathcal{F}(\mathbf{z}^{(1)})$ is finite. From the \mathcal{F} -monotonicity of the sequence $\{\mathbf{z}^{(t)}\}_{t=1}^\infty$, as asserted by theorem 4.9, and the positivity constraints incorporated into the algorithm, the iterates $\{\mathbf{z}^{(k)}\}_{k=1}^\infty$ are contained in

⁷Please see footnote 4 on page 6.

a sub-level set of \mathcal{F} given by $\{z : z \succeq 0, \mathcal{F}(z) \leq \mathcal{F}^{(1)}\}$. To show that this set is compact we only need to show that $\lim_{\|z\|_2 \rightarrow \infty} \mathcal{F}(z) = \infty$. Using (2.12)–(2.15), it can be verified that this condition is satisfied for any combination of $m_j \rightarrow \infty$, $v_j \rightarrow \infty$, $\gamma_k \rightarrow \infty$.

(b) Follows from (4.17).

(c) Equivalently, we show that if $\mathcal{F}^{(t)} \leq \mathcal{F}^{(t+1)}$ then $z^{(t)} \in \Gamma$. Assume that $\mathcal{F}^{(t)} \leq \mathcal{F}^{(t+1)}$, then from (4.17) we must have $\mathcal{F}^{(t)} = \mathcal{F}^{(t+1)}$ and all I divergences in (4.17) must equal zero (as they are nonnegative). This implies that $\beta_j^{(t)} \rightarrow b_j^{(t)}$ and $\tilde{\beta}_j^{(t)} \rightarrow \tilde{b}_j^{(t)}$, $\forall j$. From (4.11) and (4.13) it follows that $m_j^{(t+1)} \rightarrow m_j^{(t)}$, and $v_j^{(t+1)} \rightarrow v_j^{(t)}$ so $z^{(t)}$ is a fixed point. It then follows from corollary 4.8 that $z^{(t)}$ satisfies the KKT conditions, i.e., $z^{(t)} \in \Gamma$.

(d) We use a result by Gunawardana and Byrne [36] which is restated here for convenience.

Proposition 4.12 (Proposition 7 in [36]). *Given a real-valued continuous function f on $A \times B$, define the point-to-set map $F : A \rightarrow B$ by*

$$(4.19) \quad F(a) = \arg \min_{b \in B} f(a, b) = \{b : f(a, b) \leq f(a, b'), \forall b' \in B\}.$$

Then, the point-to-set mapping $a \rightarrow F(a)$ is closed at $a = a'$ if $F(a')$ is nonempty.

The surrogate functions in (3.17)–(3.18) are continuous and convex, and they also satisfy $\mathcal{S}_j^m \geq 0$ and $\mathcal{S}_j^v \geq 0$ for any $m_j, v_j \geq 0$, thus insuring the existence of the solutions to (3.19)–(3.20). From Proposition 4.12 it follows that the mapping $(\mathbf{m}^{(t)}, \mathbf{v}^{(t)}) \rightarrow (\mathbf{m}^{(t+1)}, \mathbf{v}^{(t+1)})$ defined by (3.19)–(3.20) (lines 11-12 of Algorithm 1) is a *closed* point-to-set mapping. Performing several iterations of (3.19)–(3.20) also constitutes a closed mapping, since it is a composition of closed mappings. The mapping $\gamma^{(t)} \rightarrow \gamma^{(t+1)}$ defined by (2.16) (line 15 of Algorithm 1) is a continuous and therefore closed point-to-point mapping. Finally, the composition of both closed mappings $(\mathbf{m}^{(t)}, \mathbf{v}^{(t)}, \gamma^{(t)}) \rightarrow (\mathbf{m}^{(t+1)}, \mathbf{v}^{(t+1)}, \gamma^{(t+1)})$ is also closed.

(e) Follows from (a)–(d) and Zangwill’s generalized convergence theorem [43]. ■

5. Different Perspectives on VARD and Connections to Prior Art.

5.1. A Different View of VARD. We present an alternative view of the proposed VARD framework, which will provide further insight into the sparsity-promoting mechanism and will later enable extensions of the AM algorithm. The minimum of the objective function in (2.13)–(2.15) with respect to γ is given by

$$(5.1) \quad \gamma_k^* = \left(\sum_j \psi_{kj} m_j \right)^2 + \sum_j \psi_{kj}^2 v_j.$$

Substituting (5.1) into (2.13)–(2.15) we obtain an alternative formulation for VARD

$$(5.2) \quad \min_{\mathbf{m} \succeq 0, \mathbf{v} \succeq 0} \mathbb{E}_{q(\mathbf{x}|\mathbf{m}, \mathbf{v})} [-\log p(\mathbf{y}|\mathbf{x})] + \frac{1}{2} \sum_k \log(\mu_k^2 + \sigma_k^2) - h[q(\mathbf{x}; \mathbf{m}, \mathbf{v})],$$

$$(5.3) \quad \mu_k = \sum_j \psi_{kj} m_j, \quad \sigma_k^2 = \sum_j \psi_{kj}^2 v_j, \quad h[q(\mathbf{x}; \mathbf{m}, \mathbf{v})] = \frac{1}{2} \sum_j \log v_j,$$

where the explicit expression for $\mathbb{E}_{q(\mathbf{x}|\mathbf{m}, \mathbf{v})} [-\log p(\mathbf{y}|\mathbf{x})]$ is given by (2.13), and $h[q(\mathbf{x}; \mathbf{m}, \mathbf{v})]$ in (5.3) denotes the differential entropy of the distribution q . $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ in (5.3) are the posterior mean and standard deviation (std) in the transform domain $\mathbf{s} = \boldsymbol{\Psi}\mathbf{m}$, respectively.

Note in (5.2) that the term $\sum_k \log(\mu_k^2 + \sigma_k^2)$ promotes sparsity of both the posterior mean and standard deviation in the transform domain. In fact, without the entropy term $h[q] = \sum_j \log v_j/2$ in (5.2), any $(\boldsymbol{\mu}, \boldsymbol{\sigma})$ with at least one pair of components $(\mu_k, \sigma_k) = (0, 0)$ would be a global minimum of (5.2), with 2^p possible combinations. The entropy term serves as a barrier preventing solutions with $v_j = 0$, thus avoiding these unwanted minima. Note that the concave term $\sum_k \log(\mu_k^2 + \sigma_k^2)$ in (5.2) is not separable with respect to the components of \mathbf{m} and \mathbf{v} , and the convex decomposition approach used in Sec. 3 is not applicable here.

5.2. Connections to Reweighted ℓ_2 Algorithms. The ARD approach is related to the reweighted ℓ_2 algorithm [44] and the comparison between the two for Gaussian likelihoods has been presented in [45]. It is natural to consider the reweighted ℓ_2 algorithm for the Poisson likelihood as well, which consists of the following two steps

$$(5.4) \quad \mathbf{x}^{(t+1)} = \arg \min_{\mathbf{x} \succeq 0} -\log p(\mathbf{y}|\mathbf{x}) + \frac{1}{2} \sum_k (\boldsymbol{\Psi} \mathbf{x})_k^2 / \gamma_k^{(t)},$$

$$(5.5) \quad \gamma_k^{(t+1)} = (\boldsymbol{\Psi} \mathbf{x}^{(t+1)})_k^2 + \epsilon,$$

where $\epsilon > 0$ is a fixed parameter that needs to be tuned. Note that we cannot choose $\epsilon = 0$ since some entries in $\boldsymbol{\Psi} \mathbf{x}$ are zero and this will result in infinite weights for the penalty in (5.4). For an easy comparison of (5.4)–(5.5) to the VARD algorithm in (2.10)–(2.11), we provide here again the update equations for VARD with the posterior variance fixed

$$(5.6) \quad \mathbf{m}^{(t+1)} = \min_{\mathbf{m} \succeq 0} \mathbb{E}_{q(\mathbf{x}|\mathbf{m}, \mathbf{v})} [-\log p(\mathbf{y}|\mathbf{x})] + \frac{1}{2} \sum_k (\boldsymbol{\Psi} \mathbf{m})_k^2 / \gamma_k^{(t)},$$

$$(5.7) \quad \gamma_k^{(t+1)} = (\boldsymbol{\Psi} \mathbf{m}^{(t+1)})_k^2 + (\boldsymbol{\Psi} \text{Diag}[\mathbf{v}^{(t+1)}] \boldsymbol{\Psi}^T)_k.$$

It can be seen that in (5.4) the data-fidelity term is the negated log-likelihood, whereas for VARD in (5.6) it is the expectation of this term with respect to the approximate posterior q . A key difference between the two algorithms is that in (5.5) a tuning parameter ϵ is required, whereas in (5.7) the variances serve as “tuning parameters” with one parameter for each pixel/voxel, and are automatically learned from the data during the B-step.

Next, we make another comparison between the two approaches. We rewrite the reweighted ℓ_2 algorithm in (5.4)–(5.5) as $\mathbf{x}^{(t+1)} = \min_{\mathbf{x} \succeq 0} Q(\mathbf{x}, \boldsymbol{\gamma}^{(t)})$ and $\boldsymbol{\gamma}^{(t+1)} = \min_{\boldsymbol{\gamma}} Q(\mathbf{x}^{(t+1)}, \boldsymbol{\gamma})$, where Q is defined as

$$(5.8) \quad Q(\mathbf{x}, \boldsymbol{\gamma}) = -\log p(\mathbf{y}|\mathbf{x}) + \frac{1}{2} \sum_k [(\boldsymbol{\Psi} \mathbf{x})_k^2 + \epsilon] / \gamma_k + \frac{1}{2} \sum_k \log \gamma_k.$$

Minimizing (5.8) with respect to $\boldsymbol{\gamma}$ by solving $\nabla_{\boldsymbol{\gamma}} Q = 0$ results in (5.5) with the superscript $(t+1)$ removed. Substituting this solution into (5.8), we obtain the equivalent MAP estimation problem

$$(5.9) \quad \min_{\mathbf{x} \succeq 0} -\log p(\mathbf{y}|\mathbf{x}) + \frac{1}{2} \sum_k \log [(\boldsymbol{\Psi} \mathbf{x})_k^2 + \epsilon].$$

The result in (5.9) can be compared to the alternative formulation for VARD given in (5.2)–(5.3). Note that small values of ϵ in (5.9) introduce many local minima, since any solution with at least one component $(\Psi \mathbf{x})_k = 0$ will result in a large negative penalty. In this case, any standard gradient-based algorithm will be easily trapped at a sub-optimal solution. On the other hand, large values of ϵ avoid local minima but do not promote sparse solutions. The correct choice for the parameter ϵ is therefore critical.

To facilitate the comparison between reweighted ℓ_2 and PAR-VARD, we propose to modify the former using the separable surrogate approach as in Sec. 3, and derive the corresponding surrogate function. The surrogate function for the log-likelihood is given by

$$(5.10) \quad -\log p(\mathbf{y}|\mathbf{x}) \leq \sum_j \mathcal{S}_j^{mle}(x_j; x_j^{(t)}) \triangleq \sum_j \left\{ b_j^y x_j + \frac{b_j^{(t)}}{Z_1} \exp[-Z_{mle}(x_j - x_j^{(t)})] \right\},$$

where $b_j^{(t)}$ and b_j^y are computed according to (3.5) and (3.4) with $\hat{p}_j^{(t)} = 0$ plugged in, and

$$(5.11) \quad Z_{mle} = \max_i \sum_j \phi_{ij}.$$

$\mathcal{S}_j^{mle}(x_j)$ in (5.10) was derived in [13] and can be obtained as a special case of the PAR-VARD surrogate in (3.17) when \mathbf{m} is replaced by \mathbf{x} , $\mathbf{v} \rightarrow 0$, and $\gamma \rightarrow \infty$, with consequently $f_j^{(t)}, g_j^{(t)} \rightarrow 0$. Adding the surrogate function for the quadratic penalty in (5.4), we obtain the update equation

$$(5.12) \quad x_j^{(t+1)} = \arg \min_{x \geq 0} \mathcal{S}_j^x(x; x_j^{(t)}),$$

$$(5.13) \quad \mathcal{S}_j^x(x; x_j^{(t)}) = \mathcal{S}_j^{mle}(x; x_j^{(t)}) + f_j^{(t)}(x - x_j^{(t)}) + g_j^{(t)}(x - x_j^{(t)})^2,$$

where $f_j^{(t)}, g_j^{(t)}$ are defined in (3.13)–(3.14) with $\gamma_j^{(t)}$ given by (5.5) and $m_j^{(t)}$ replaced by $x_j^{(t)}$. The reweighted ℓ_2 algorithm is then described in Algorithm 2. To the best of our knowledge, this algorithm has not been previously reported in the literature and can be considered as another contribution of this paper.

5.3. Comparison to Maximum Likelihood Estimation (MLE) and Maximum a Posteriori (MAP) Estimation. Next, we compare VARD to the AM algorithm for maximum likelihood estimation (MLE) in [13] and the AM algorithm for penalized maximum likelihood estimation (MAP) in [47, 48]. The model in [13] is beyond the scope of our work, but it can be reduced to the special case of a monoenergetic model with no background events given in (1.1) (See also Sec. 7 for a more detailed discussion). Although the derivations in this section can be found in [13, 47, 48] we present them here to allow for an easier comparison to the proposed VARD algorithm.

The surrogate function for MLE is given by (5.10). The minimizer of this surrogate is given in closed form, leading to the update equation

$$(5.14) \quad x_j^{(t+1)} = \left[x_j^{(t)} + \frac{1}{Z_{mle}} \log \left(\frac{b_j^{(t)}}{b_j^y} \right) \right]_+,$$

Algorithm 2 : Reweighted ℓ_2 with Separable Surrogates

```

1: Initialize  $\mathbf{x}^{(1)}, \gamma^{(1)}$ 
2: Compute  $\mathbf{b}^y$  in (3.5) % backprojection of data %
3: for t=1 to N do % AM iterations %
4:    $\mathbf{p}^{(t)} \leftarrow \Phi \mathbf{x}^{(t)}$  % projections %
5:    $\boldsymbol{\mu}^{(t)} \leftarrow \boldsymbol{\eta} \odot \exp(-\mathbf{p}^{(t)})$ 
6:   Compute  $\mathbf{g}^{(t)}$ , and  $\mathbf{f}^{(t)}$  % defined in (3.13)–(3.14) with  $m^{(t)}$  replaced by  $x^{(t)}$  %
7:    $\mathbf{b}^{(t)} \leftarrow \Phi^T \boldsymbol{\mu}^{(t)}$  % backprojections of  $\boldsymbol{\mu}^{(t)}$  %
8:   for j=1 to p do % executed in parallel %
9:      $x_j^{(t+1)} \leftarrow \arg \min_{x \geq 0} \mathcal{S}^x(x; x_j^{(t)}, b_j^{(t)}, f_j^{(t)}, g_j^{(t)})$  % see (5.12)–(5.13) %
10:   $\gamma^{(t+1)} \leftarrow [\Psi \mathbf{x}^{(t)}] \odot [\Psi \mathbf{x}^{(t)}] + \mathbf{1}\epsilon$  %  $\mathbf{1}$  is the vector of all ones %

```

Algorithm 3 : MLE/ MAP with Separable Surrogates

```

1: Initialize  $\mathbf{x}^{(1)}$ 
2: Compute  $\mathbf{b}^y$  in (3.5) % backprojection of data %
3: for t=1 to L do % convex decomposition iterations %
4:    $\mathbf{p}^{(t)} \leftarrow \Phi \mathbf{x}^{(t)}$  % projections %
5:    $\boldsymbol{\mu}^{(t)} \leftarrow \boldsymbol{\eta} \odot \exp(-\mathbf{p}^{(t)})$  % predicted mean for measurements %
6:    $\mathbf{b}^{(t)} \leftarrow \Phi^T \boldsymbol{\mu}^{(t)}$  % backprojections of  $\boldsymbol{\mu}^{(t)}$  %
7:   for j=1 to p do % executed in parallel %
8:      $x_j^{(t+1)} = [x_j^{(t)} + \log(b_j^{(t)}/b_j^y)/Z_{mle}]_+$  % only for MLE %
9:     or  $x_j^{(t+1)} \leftarrow \arg \min_{x \geq 0} \mathcal{S}^{map}(x; x_j^{(t)}, b_j^{(t)})$  % only for MAP (see (5.17)) %

```

with $b_j^{(t)}$ and b_j^y computed according to (3.5) and (3.4) with $\tilde{p}_j^{(t)} = 0$ plugged in, and Z_{mle} is defined in (5.11). The resulting algorithm is described in Algorithm 3.

Using the same approach, it is also possible to add a penalty to the log-likelihood and obtain an AM algorithm for MAP estimation [47, 48]. In the following comparisons, we shall use the smooth edge preserving neighborhood penalty given by (1.4)–(1.5). The surrogate function for this penalty involving the j th pixel is given by

$$(5.15) \quad \mathcal{S}_j^{pen}(x; x_j^{(t)}) = \beta \delta^2 \sum_{k: \{j,k\} \in \Theta} \frac{1}{N_j} \left[|x_j^{(t)} - x_k^{(t)} + N_j(x - x_j^{(t)})|/\delta - \log(|x_j^{(t)} - x_k^{(t)} + N_j(x - x_j^{(t)})|/\delta) \right],$$

where N_j is the number of neighbors of the j th pixel. The updates for the MAP estimate are then given by

$$(5.16) \quad x_j^{(t+1)} = \arg \min_{x \geq 0} \mathcal{S}_j^{map}(x; x_j^{(t)}),$$

$$(5.17) \quad \mathcal{S}_j^{map}(x; x_j^{(t)}) = \mathcal{S}_j^{mle}(x; x_j^{(t)}) + \mathcal{S}_j^{pen}(x; x_j^{(t)}),$$

where \mathcal{S}_j^{mle} and \mathcal{S}_j^{pen} are given by (5.10) and (5.15), respectively. See Algorithm 3.

5.4. Connections to Previous Variational Inference Methods. The approximation of the posterior using a Gaussian distribution has been studied by Challis and Barber [49]. Although they did not consider the Poisson likelihood in (1.1) specifically, their approach is the same as our B-step in (2.10) if the posterior is chosen to be factorized and if γ are *fixed* and *known*. We emphasize that this is very different from the AM algorithm proposed here, which alternates between the B-step and the F-step where γ are also updated. In fact, a single B-step does not result in sparse solutions. In order to obtain sparse solutions, the B and F steps need to be repeated. This is similar in principle to the original EM-ARD algorithm [15], where a single E-step would not result in sparse solutions. Also, the parallel algorithm we propose in Sec. 3 is completely new. An alternative approach to [49] was proposed by Khan et al. [50] but it requires the inversion of a $p \times p$ matrix (p is the number of pixels/voxels) so it is not feasible for large scale problems. We note that the method by Seeger and Nickisch [30] cannot be applied here for approximating the posterior since it is restricted to super-Gaussian likelihoods and the Poisson likelihood we consider (see (1.1)) is not super-Gaussian.

6. Generalization of VARD to Improper Priors and Overcomplete Representations.

Building on Sec. 5.1, we extend VARD to cases when the prior is improper or when one wishes to use a non-square Ψ matrix in (2.1). Recall that when deriving the AM algorithm in Sec. 2.3, we assumed a proper prior (the matrix $\Psi^T \Gamma^{-1} \Psi$ is non-singular) and that Ψ is a non-singular square matrix (see footnote 5). These assumptions are violated when the Dirichlet boundary conditions in Sec. 2.1 cannot be used. For example, in cone beam CT, the object extends beyond the region of interest along one spatial direction, and the Dirichlet boundary conditions are violated in that direction. It is well known that without boundary conditions, the neighborhood penalties of the form in (1.4) correspond to a singular Ψ matrix in (2.1) [51], leading to an improper prior. The case when Ψ is non-square arises when one wishes to use an overcomplete sparse representation, which is shown in Sec. 9 to yield better image quality. Even if Ψ is chosen such that the prior is proper, introducing a non-square Ψ in the AM algorithm of (2.10)–(2.11) results in a considerable complication of the γ update (F-step). The objective includes $\log |(\Psi^T \Gamma^{-1} \Psi)^{-1}|$ which can no longer be reduced to $\sum_k \log \gamma_k$ ($\Gamma = \text{Diag}(\gamma)$) if Ψ is non-square. Taking the gradient of this expression with respect to γ yields $\gamma^{-2} \odot \text{Diag}[\Psi(\Psi^T \Gamma^{-1} \Psi)^{-1} \Psi^T]$. Both the objective and the gradient have a computational complexity of $O(p^3)$ (p = number of pixels/voxels) which is prohibitive for large-scale problems.

To avoid the above complications, we use a different approach, where we start with the objective function in (5.2)–(5.3), regardless of how it was derived, where Ψ can be square but singular or non-square. This can be justified by the insights gained from Secs. 5.1–5.2. The objective in (5.2)–(5.3) can be minimized by using an AM algorithm for minimizing the objective function in (2.12)–(2.15). The latter objective can no longer be interpreted as the free variational energy (FVE) in (2.5), even when the prior is proper, since we replaced $\log |(\Psi^T \Gamma^{-1} \Psi)^{-1}|$ with $\sum_k \log \gamma_k$. Since the replaced term is independent of \mathbf{m} and \mathbf{v} , minimizing the objective in (2.12)–(2.15) with respect to (\mathbf{m}, \mathbf{v}) during the B-step of the AM algorithm is still equivalent to minimizing the KL divergence between the proposed posterior q and the true posterior⁸. The F-step, on the other hand, can no longer be interpreted

⁸Note that the posterior distribution defined by (1.3) can be well-defined even though the prior is improper.

as minimizing the KL divergence between q and the prior, as in (2.11). Nevertheless, the promotion of sparsity is still asserted by the equivalent formulation in (5.2)–(5.3), where the $\sum_k \log(\mu_k^2 + \sigma_k^2)$ term promotes sparsity in the transform domain, and it is also supported by numerical experiments in Sec. 9. It is important to note that although the interpretation of the objective function during the F-step is different, it is still the same objective as before, given by (2.12)–(2.15), so the *convergence analysis in Sec. 4 still applies*.

To illustrate the type of priors that the above extension allows, we give an example for 2D-images where Ψ is a difference matrix given by

$$(6.1) \quad \Psi = [\Psi^h; \Psi^v], \quad (\Psi^{h,v})_{jk} = \begin{cases} 1 & \text{if } j = k \\ -1 & \text{if } \{j, k\} \in \Theta^{h,v} \\ 0 & \text{otherwise} \end{cases},$$

where $[\cdot]$ denotes row-wise concatenation, and Θ^h, Θ^v define pairs of neighboring pixels located along the same horizontal or vertical lines in the image, with corresponding matrices Ψ^h, Ψ^v , respectively. We assign the same weight to horizontal and vertical pixel-differences, so γ is replaced by $[\gamma; \gamma]$. It is also possible to incorporate differences between additional neighbors by concatenating additional Ψ matrices for these additional neighbors in (6.1), and it is also easily extendible to 3D images. To understand the difference between the above construction and the original one in (2.3), we write the sparsifying penalty from the objective function in (5.2) for both cases with only 2 neighbors per pixel for simplicity

$$(6.2) \quad \mathcal{P}_1 = \sum_{j=1}^p \log \left\{ \sqrt{\left(m_j - \frac{(m_{j_1} + m_{j_2})}{2}\right)^2 + v_j + \frac{1}{4}v_{j_1} + \frac{1}{4}v_{j_2}} \right\},$$

$$(6.3) \quad \mathcal{P}_2 = \sum_{j=1}^p \log \left\{ \sqrt{(m_j - m_{j_1})^2 + (v_j + v_{j_1}) + (m_j - m_{j_2})^2 + (v_j + v_{j_2})} \right\},$$

where \mathcal{P}_1 corresponds to the original construction in (2.3) and \mathcal{P}_2 corresponds to the construction in (6.1). In both cases, j_1 and j_2 denote the indexes of the horizontal and vertical neighbors of the j th pixel, respectively. For \mathcal{P}_1 we have $\{j, j_1\} \in \Theta$ and $\{j, j_2\} \in \Theta$, so both neighbors are included in the same neighborhood defined by Θ . For \mathcal{P}_2 we have $\{j, j_1\} \in \Theta^h$ and $\{j, j_2\} \in \Theta^v$, using two different neighborhoods. In the limit of the variances going to zero, (6.3) reduces to the logarithm of the isotropic total variation (TV) penalty, i.e., \mathcal{P}_2 promotes small differences between neighbors equally in each direction. In contrast, using the same limit in the argument of the logarithmic function in (6.2) reduces it to differences between a pixel and the *average* of its neighbors, so \mathcal{P}_1 promotes piecewise smoothness.

7. Model Limitations. The algorithm presented in this paper is designed to solve a specific ARD problem. As with all models, the model assumed in Sec. 1.1 is only approximate, capturing some but not all of the underlying physics. Considering a simplified physical model also facilitated our theoretical study which could potentially apply more broadly to applications other than the one we consider to motivate our work. Next, we discuss the limitations of the forward model presented in Sec. 1.1 when applying it to x-ray CT.

The actual data in X-ray transmission tomography are typically not Poisson distributed due to either system non-linearities or fundamental physical considerations. For example,

the solid state detectors currently used by modern CT scanners are energy-integrating detectors, not photon-counting detectors. The signal statistics for energy-integrating detectors are studied by Whiting et al. [52, 53]. As mentioned in Sec. 1, there is a growing interest in photon-counting detectors, which are already used in clinical studies [18], and it is possible that this type of detectors will be integrated into commercial CT systems in the near future. Various researchers have developed algorithms for models that account for various physical phenomena such as the poly-energetic nature of the source [13], background events due to scatter [12, 13], and electronic noise, which are beyond the scope of this work. A review on modelling the physics in iterative algorithms can be found in Nuyts et al. [54].

Another simplification we have used is in the construction of the system matrix Φ which was based on line length intersections. This approach does not take into account the detector pixel size or the focal spot size of the source. More realistic approaches for computing the system matrix can be found in the work by Basu and DeMan [55, 56].

Finally, we would like to point to the limitations of the specific priors presented in Sec. 2.1 and Sec. 6 which use the Ψ matrix defined in (2.3) and (6.1), respectively. As mentioned in Sec. 6, in the case of cone-beam CT, the object often extends beyond the region of interest in one spatial direction, so the Dirichlet boundary conditions assumed implicitly in (2.3) and (6.1) cannot be applied in this direction. In this case, the extension of VARD presented in Sec. 6 can be used to choose a different Ψ matrix which does not implicitly assume any boundary conditions.

8. Alternative Algorithms. In developing the parallel AM algorithm of Sec. 3 we adopted a particular optimization transfer technique for the B-step based on separable surrogates that were derived via the convex decomposition lemma (see lemma 3.1). There are several alternative approaches in the literature for CT reconstruction that could be used for the B-step as well. Two important examples are the separable paraboloidal surrogates (SPS) algorithm [12] and the iterative coordinate descent (ICD) algorithm [19, 57]. Each different choice of an optimization method for the B-step would result in a different VARD algorithm, which could be compared to the corresponding MAP algorithm using the same optimization method. Here we consider only one possible choice and exploring the others is left for future research.

9. Numerical Results. In the following section we study the performance of the proposed VARD algorithm for x-ray CT. We consider both simulated and real data for x-ray CT with source intensity levels (η values in (1.1)) which are typical of clinical settings. The geometry considered in the experiments is shown in Fig. 4.

We compare the following algorithms: (1) The proposed PAR-VARD algorithm described in Algorithm 1; (2) Maximum likelihood estimator (MLE) from [13] described in Algorithm 3; (3) Maximum a Posteriori (MAP) estimator from [47, 48] described in Algorithm 3; (4) The reweighted ℓ_2 algorithm described in Algorithm 2. For simplicity, we shall slightly abuse notation and refer to PAR-VARD as VARD. For reweighted ℓ_2 and VARD, we use the prior in (2.1) with two choices for Ψ given by (2.3) and (6.1), which we refer to as “complete” (C) and “over-complete” (O) representations, respectively. In (2.3) we choose Θ to correspond to a neighborhood of 1 horizontal and 1 vertical neighbors. For MAP, we use a sum of two penalties constructed by using (1.4)–(1.5), one for the horizontal neighbor and one for the vertical neighbor with Dirichlet boundary conditions (consistent with (2.3) and (6.1)). All

| (a) MAP | | | | | (b) MLE/VARD | |
|----------------|--------------------|--------------------|--------------------|--------------------|---------------|---------------|
| | $\delta = 10^{-5}$ | $\delta = 10^{-4}$ | $\delta = 10^{-3}$ | $\delta = 10^{-2}$ | MLE | 1.63 % |
| $\beta = 10^5$ | 1.79 % | 1.52 % | 0.65 % | 2.10 % | VARD-C | 0.85 % |
| $\beta = 10^6$ | 1.51 % | 0.54 % | 1.16 % | 9.26 % | VARD-O | 0.68 % |
| $\beta = 10^7$ | 0.55 % | 0.96 % | 7.22 % | 40.01 % | | |
| $\beta = 10^8$ | 1.61 % | 11.5 % | 43.69 % | 64.59 % | | |

| (c) Reweighted ℓ_2 | | | | |
|-------------------------|----------------------|----------------------|----------------------|----------------------|
| | $\epsilon = 10^{-8}$ | $\epsilon = 10^{-6}$ | $\epsilon = 10^{-4}$ | $\epsilon = 10^{-2}$ |
| Reweighted ℓ_2 -C | 11.03 % | 0.66 % | 1.32 % | 1.81 % |
| Reweighted ℓ_2 -O | 2.88 % | 0.41 % | 0.99 % | 1.79 % |

Table 1

Normalized root-mean-square errors (NRMSE) for reconstructions from synthetic data with Poisson noise using different methods and different values of tuning parameters. RMSE is normalized relative to the ℓ_2 norm of the vectorized ground truth image. C=“Complete” and O=“over-complete” indicate the use of Ψ in (2.3) and (6.1), respectively. Here $\eta = 10^5$. The standard deviation (std) of the NRMSE over 50 different noise realizations was negligible and therefore was not taken into account, e.g., for VARD-O, $\text{std} \approx 0.015\%$.

| (a) MAP | | | | | (b) MLE/VARD | |
|----------------|--------------------|--------------------|--------------------|--------------------|---------------|---------------|
| | $\delta = 10^{-5}$ | $\delta = 10^{-4}$ | $\delta = 10^{-3}$ | $\delta = 10^{-2}$ | MLE | 5.64 % |
| $\beta = 10^5$ | 5.29 % | 3.01 % | 1.51 % | 9.30 % | VARD-C | 2.45 % |
| $\beta = 10^6$ | 2.99 % | 1.36 % | 7.27 % | 40.00 % | VARD-O | 1.76 % |
| $\beta = 10^7$ | 1.85 % | 11.60 % | 43.69 % | 64.59 % | | |
| $\beta = 10^8$ | 21.94 % | 58.19 % | 73.94 % | 76.32 % | | |

| (c) Reweighted ℓ_2 | | | | |
|-------------------------|----------------------|----------------------|----------------------|----------------------|
| | $\epsilon = 10^{-8}$ | $\epsilon = 10^{-6}$ | $\epsilon = 10^{-4}$ | $\epsilon = 10^{-2}$ |
| Reweighted ℓ_2 -C | 46.2 % | 3.54 % | 2.49 % | 5.23 % |
| Reweighted ℓ_2 -O | 21.34 % | 1.5 % | 1.16 % | 4.80 % |

Table 2

Same as Table 1 but for $\eta = 10^4$.

which is the best choice we could find, and the other is a bad choice. One can see in Figs. 5(d) and 5(f) how significant artifacts occur when the tuning parameters are not chosen correctly. In contrast, the VARD reconstruction in Fig. 5(g) does not have any significant artifacts and the image quality is comparable to the best results obtained with the alternative methods (Figs. 5(c), 5(e)). The reconstructed hyperparameters γ are shown in Fig. 6(a) where one can see that γ has large values at discontinuities and very low values in regions where the attenuation levels are constant. Recall that VARD with the prior in (6.1) promotes sparsity in the pixel-difference domain. Figure 6(a) clearly indicates that VARD has learned the correct support of the object in the pixel-differences domain. The posterior standard deviation (std) is shown in Fig. 6(b) and has a similar structure to γ . Note that one should be careful when

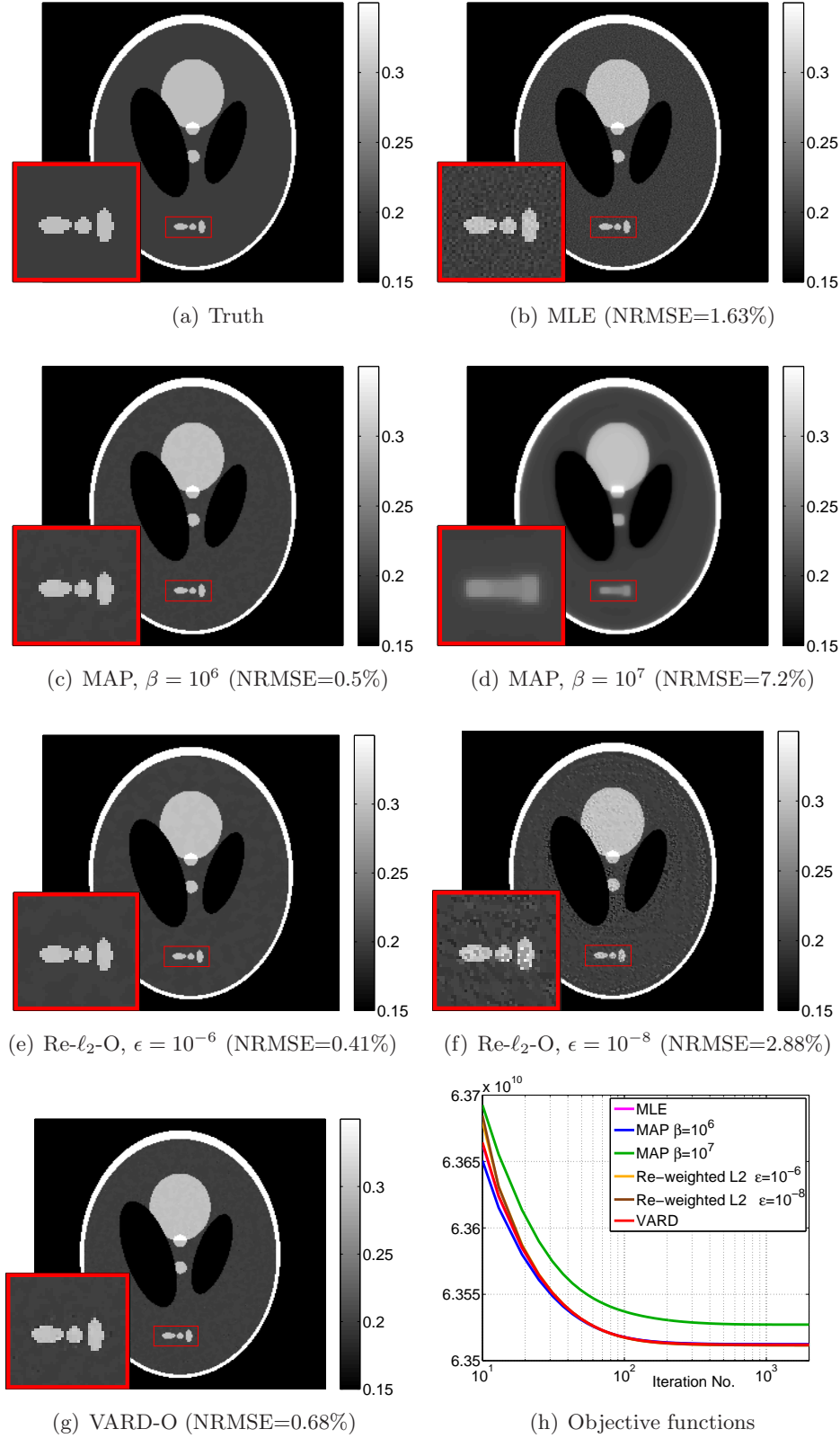


Figure 5. Reconstructions of attenuation images (relative to water) from synthetic data with Poisson noise using different methods. All images have resolution 256×256 . (a) Truth; (b) Maximum likelihood (MLE) estimate; (c) Maximum a posteriori (MAP) estimate using the smooth edge-preserving neighborhood penalty with $\beta = 10^6$ and $\delta = 10^{-4}$; (d) same as Fig. (c) but with $\beta = 10^7$ and $\delta = 10^{-3}$; (e) Reweighted ℓ_2 -O with $\epsilon = 10^{-6}$; (f) same as (e) but with $\epsilon = 10^{-8}$; (g) Posterior mean using VARD-O; (h) Objective functions for all methods (note that each algorithm is optimizing a different objective). All images are displayed using the window $[0.15, 0.35]$, even though the phantom includes values in $[0, 1]$, in order to enhance the differences between the different methods. A zoom-in view of the region inside the small red frame is shown in the inset to the left in figures (a)–(g). All algorithms except MLE use a neighborhood penalty with one horizontal and one vertical neighbor per pixel. All algorithms were run till 2000 iterations to ensure convergence. We recommend viewing the figures with at least 150% zoom-in.

interpreting the posterior std since the support of the posterior is $(-\infty, \infty)$ (see (2.9)) and one std from the mean can extend to negative attenuation values. In Sec. 9.2 we show a useful interpretation of the posterior variances.

Figure 5(h) presents the objective function for VARD-O ((2.12)–(2.15)) vs. number of iterations, as well as the objectives for all the other methods. The objective is seen to decrease monotonically, as predicted by theory, and can be used to assess convergence in practice, as well as to verify the correctness of the implementation. It can also be seen that all methods share similar convergence rates. It is important to note that each algorithm is optimizing a *different* objective function so to compare convergence rates one needs to compare the slopes of the objectives and not the objective values.

In Table 3 we compare VARD to the ARD method from [17] (SBL) which is based on an approximate post-log Gaussian noise model. Details about SBL can be found in Appendix C. After convergence, we set all the negative pixel values in the SBL image to zero in order to obtain a physically meaningful result (since SBL does not impose positivity). From Table 3 one can see that SBL leads to higher NRMSE than VARD-C. For the highest photon flux (lowest noise level) with $\eta = 10^5$, SBL [17] is comparable to VARD-C, but as the photon flux gets lower ($\eta = 10^4$ and $\eta = 10^3$), the difference between SBL and VARD-C becomes more noticeable. Since SBL and VARD-C use the same prior, the results suggest that the approximate Gaussian noise model is quite effective for the highest photon flux but becomes less effective as the flux decreases, as expected. Importantly, SBL has a notably higher RSME than VARD-O, especially for lower flux levels, which suggests that the over-complete prior in (6.1) used by VARD-O provides another advantage over SBL. Note that SBL does not allow the use of the over-complete prior due to the assumptions used in deriving the M-step [17], which are similar to the ones used for the F-step in Sec. 2.3 but were later lifted in Sec. 6. We initialized SBL in the same way as VARD, and SBL took 100 iterations to converge. Note that each iteration of SBL has a much higher computational cost than an iteration of VARD (see details in Sec. 9.4). To illustrate the differences in computational costs, we note that VARD took about 20 minutes to run whereas SBL took 5 hours! (SBL was run on CPU's, like VARD, and not on a GPU as in [17]).

| | $\eta = 10^5$ | $\eta = 10^4$ | $\eta = 10^3$ |
|--------|---------------|---------------|---------------|
| SBL | 0.97 % | 3.15 % | 9.82 % |
| VARD-C | 0.85 % | 2.45 % | 7.35 % |
| VARD-O | 0.68 % | 1.76 % | 5.2 % |

Table 3

Comparison of the NRMSE obtained using VARD and SBL [17] for different photon flux levels.

There are several additional advantages to using VARD instead of SBL: (1) SBL does not enforce positivity of the solution, whereas in VARD, it is built into the algorithm. Note that setting all the negative pixel values to zero after SBL converges might be sub-optimal. (2) Generally, the straightforward computation of the posterior variances requires $O(p^3)$ operations and $O(p^2)$ of memory (p is the number of pixels/voxels) which is infeasible for large scale problems so approximations are needed [17]. However, these approximations result in the loss of convergence guarantees for the overall SBL algorithm. In contrast, VARD assumes

a posterior distribution that while inexact naturally leads to lower computational costs and guarantees convergence. (3) In SBL, one is required to choose the type and number of probing vectors for approximating the variances, which are system and object dependent. (4) Computing the objective function of SBL scales as $O(p^3)$, making this operation infeasible (in this example, we could not compute the SBL objective within reasonable time), whereas computing the objective for VARD scales as $O(np^{1/D})$ with n being the number of measurements and $D = 2, 3$ for reconstruction of 2D/3D images, respectively. The objective provides another tool for checking the implementation and can be used to assess convergence.

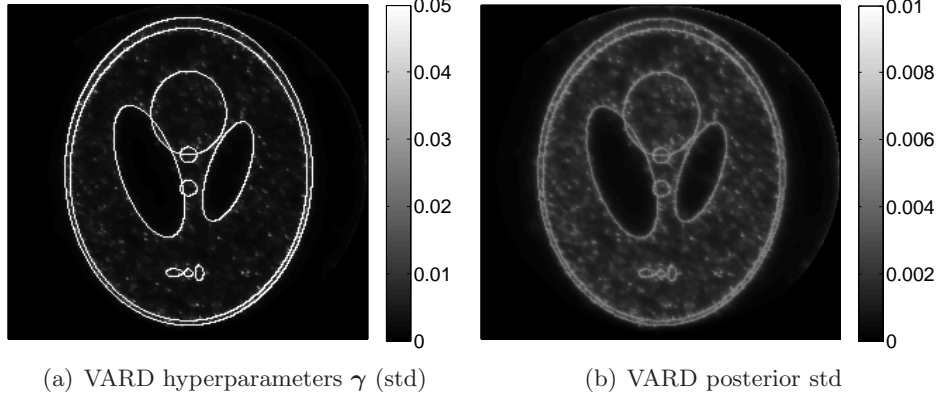


Figure 6. Reconstructed standard deviations (std) using VARD on synthetic data. (a) squared root of hyperparameters γ ; (b) squared root of posterior variances v .

9.2. Real Data. Next we present reconstructions for a dataset acquired at an experimental laboratory at Duke University with the geometry specified in Fig. 4. The number of views was 360 (full scan around the object) with view-angle resolution of 1° . For each view 1781 detector pixels were used. The total number of measurements was 641,160. Additional details about the experimental setup are provided in Appendix D. We measured a 2D cross section of an acrylic glass cylinder full of cylindrical holes with different diameters. In this case, the image for the ground truth is unknown. However, a good estimate of the truth is based on the specifications from the manufacturer of this phantom. The attenuation of the acrylic glass relative to water is 1.5, and for the inside of the rods it is 0 (air). Figure 7 presents reconstructions using different methods. Image resolution is 256×256 . For MAP and reweighted ℓ_2 we show two examples of reconstructions, one for the best choice of the tuning parameter that we could find, and one with significant artifacts for a bad choice of tuning parameters. Again, one can see how VARD produces images which are comparable in quality to the best trial with MAP or reweighted ℓ_2 but without any need for repeated trials. Figure 8 shows the reconstructed posterior variances and hyperparameters, where one can see again how VARD learns the correct support of the object in the pixel-differences domain.

Finally, in Figs. 9–10 we present an example which illustrates the robustness of VARD to missing data and also demonstrates a useful interpretation of the posterior variances. In this example we scanned a “DUKE” letters object suspended in mid-air using the same experimental system. The letters are made from VeroBlue material with a linear attenuation

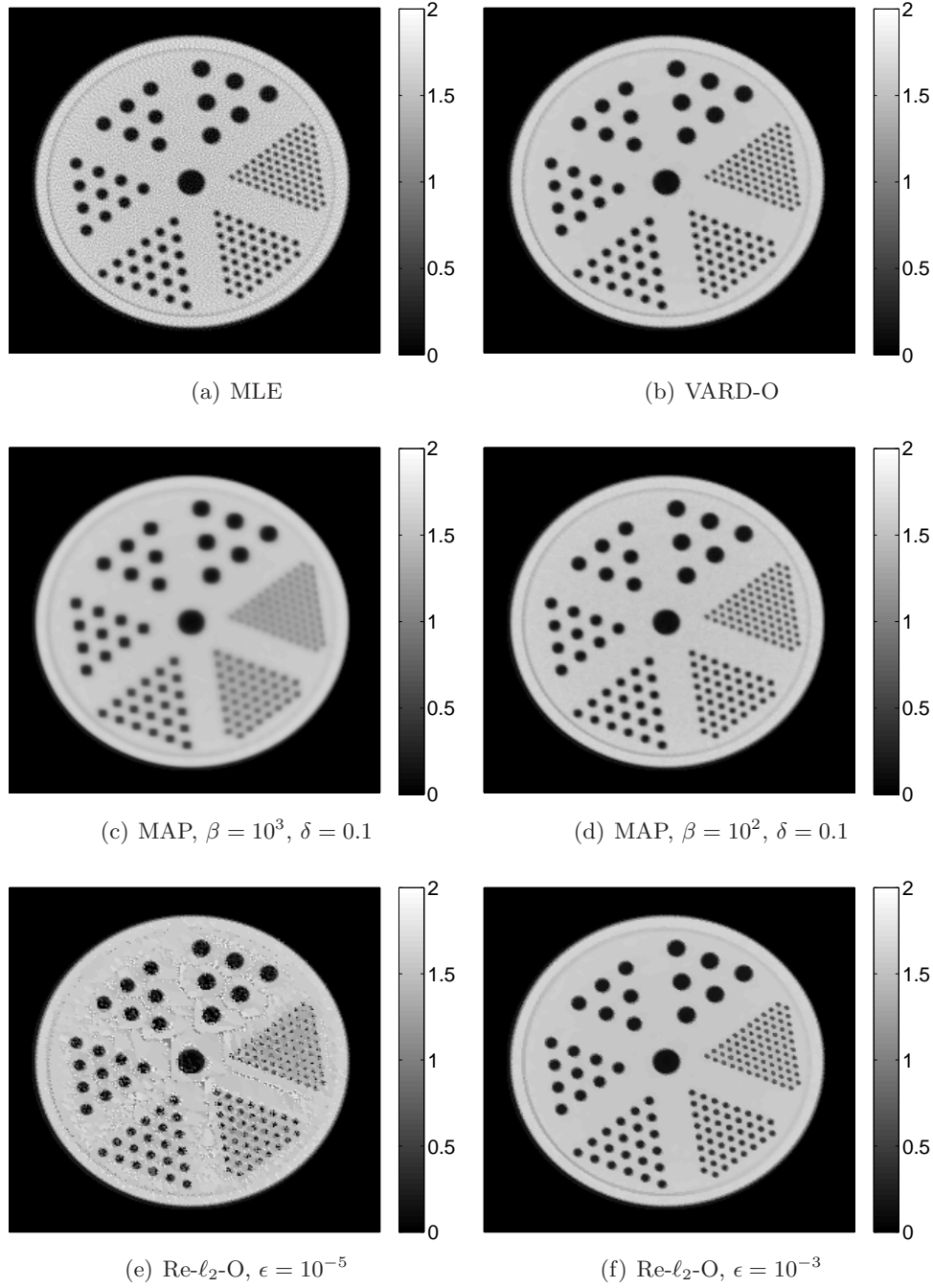


Figure 7. Reconstructions of attenuation maps (relative to water) from real data using different methods. (a) Maximum likelihood (MLE) estimate; (b) VARD-O; (c) Maximum a posteriori (MAP) estimate using the smooth edge-preserving neighborhood penalty with $\beta = 10^3$ and $\delta = 0.1$; (d) Same as (c) but with $\beta = 10^2$; (e) Reweighted ℓ_2 -O with $\epsilon = 10^{-5}$; (f) Same as (e) but with $\epsilon = 10^{-3}$. All algorithms use 2000 iterations to ensure convergence. All except MLE use a neighborhood penalty with one horizontal and one vertical neighbor per pixel. We recommend viewing the figures with at least 150% zoom-in. All objectives were monotonically decreasing and flat when the algorithms were terminated.

coefficient of 1.458 relative to water (more details in caption of Fig. 9). The expected image is sparse so we used the prior in (2.1) with $\Psi = \mathbf{I}$ to promote sparsity in the pixel basis.

We subsampled the angular grid for the source-locations (views) so only 1/8 or 1/16 of the views are used; the VARD reconstructions for these two cases are shown in Fig. 9(a) and Fig. 9(b), respectively. In addition, we include in Fig. 9(c) and 9(d) the reconstructions using filtered-back-projection (FBP) [1], which is a linear reconstruction method based on Fourier inversion. Note that the FBP reconstruction exhibits much more prominent artifacts than VARD in the form of streaks, which are due to view undersampling. We present the FBP reconstructions again in Figs 10(a)–10(b) where the displayed attenuation values were limited to a smaller range in order to enhance the artifacts. There is a striking resemblance between the streaks in the posterior std images obtained by VARD in Figs 10(c)–10(d) and the streaks in the FBP reconstructions in Figs 10(a)–10(b). This demonstrates that the variances can indicate regions in the image which are insufficiently sampled. A more detailed study of this relationship will be deferred to a subsequent publication. Lastly, we note that although we have focused on the 2D fan-beam geometry in the numerical experiments, the proposed method can be applied to any other geometry.

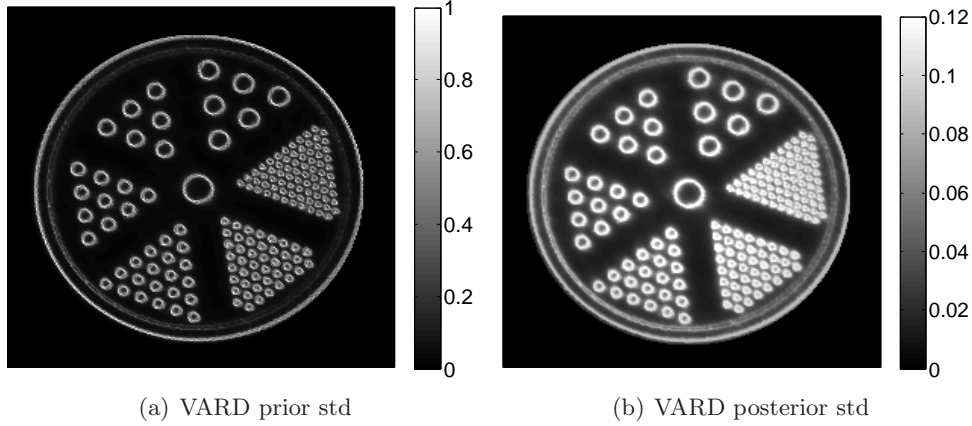


Figure 8. Reconstructed standard deviations (std) using VARD on real data. (a) squared root of hyperparameters γ ; (b) squared root of posterior variances v .

9.3. Implementation Details. In our implementation, the 1D optimizations in lines 11–12 of Algorithm 1, line 9 of Algorithm 2 and Algorithm 3 are all done in parallel using MATLAB’s vectorization. Since it is not necessary to solve the 1D surrogate problems exactly, we found that it is more efficient to use Newton’s method and perform just one Newton step for each iteration. However, for MAP this leads to divergence whenever the solution is in the linear regime of the smooth edge preserving penalty. Therefore, we use a 1D trust-region Newton method (see details in Appendix B), which typically costs about 1 – 5 Newton steps. For VARD we could use Newton steps for the mean updates, but had to resort to the trust-region method for the variance updates (see details in Appendix B) to ensure decrease of the objective. For reweighted ℓ_2 the Newton steps worked successfully. For the problem sizes considered here, we could store the system matrix and its transpose in memory using

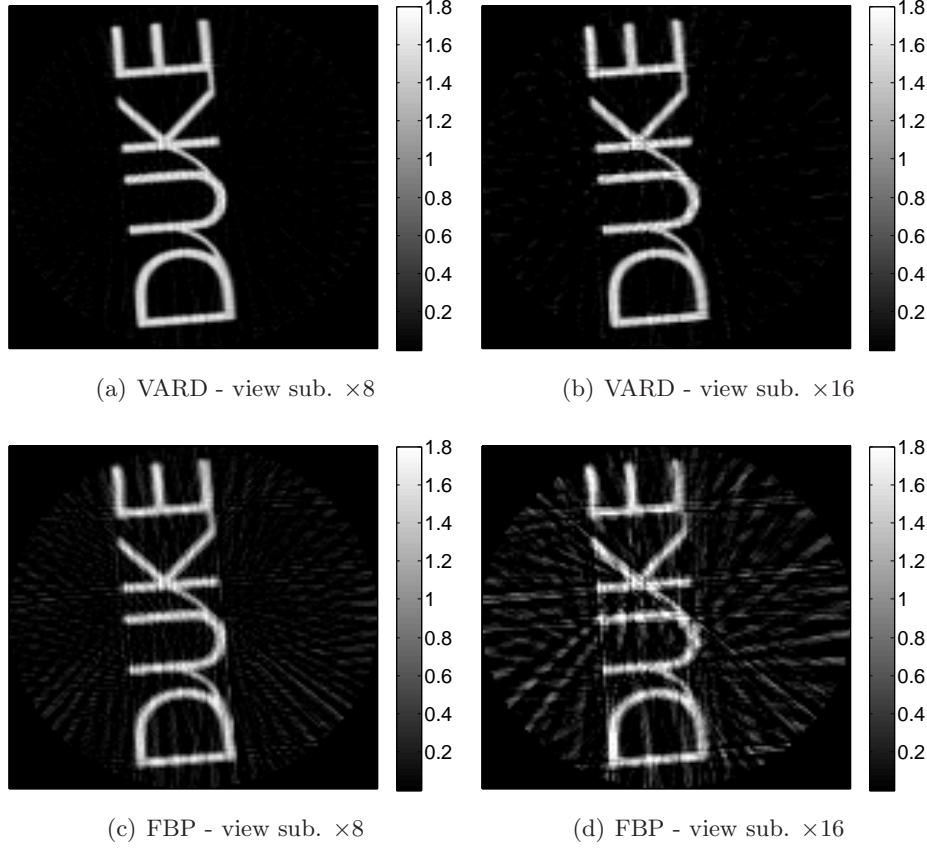


Figure 9. Reconstructions from real data. (a) VARD with sparsity promoted in the pixel basis and using 1/8 of the views; (b) same as (a) but using 1/16 of the views; (c) Filtered back projection using 1/8 of the views; (d) same as in (c) but using 1/16 of the views. For FBP, we used cubic interpolation for the backprojection, and a cropped Ram-Lak (ramp) filter. Image resolution is 128×128 . The full dataset (before the view subsampling) was acquired using a full 360° scan around the object with view-angle resolution of 1° and 1573 detector pixels per view. 510 iterations were used for VARD.

MATLAB’s compressed sparse representation. The time for the forward/back-projections was negligible relative to the trust-region Newton updates due to MATLAB’s highly optimized code for matrix-vector multiplications. However, this is certainly not a typical situation to be expected in large scale problems since it will not be possible to store the matrices in memory (even in sparse format) and “on-the-fly” implementations of the forward/back-projection operations would be required. Therefore, we do not report actual run time but instead discuss below the computational complexity per iteration for large scale problems. Details regarding the implementation of SBL can be found in Appendix C.

9.4. Computational Complexity. In analyzing the computational complexity per iteration for large scale problems it is common to consider only the forward/back-projections. It is also usually the case that these operations dominate the execution time per iteration in large scale problems. The MAP and reweighted ℓ_2 algorithms both have the same computa-

tional complexity, as they both require one forward-projection and one back-projection per iteration. To compare VARD to the other algorithms, first note that a variance-type operation (corresponding to an elementwise squared system matrix) requires double the number of multiplications relative to a mean-type operation when the projections are performed via ray-tracing and computing line intersections. If the mean-type and variance-type operations can be done in parallel then VARD requires the same number of forward/backprojections as MAP but it will have double the complexity of MAP due to the $\times 2$ larger number of multiplications in the variance-type operations. Also, if the computational resources are limited and the mean-type and variance-type operations can only be performed sequentially, then VARD will be at least 3 times more expensive per iteration than a single MAP or reweighted ℓ_2 trial. However, in cases when the tuning parameters are unknown or better ones are needed, MAP-based methods require multiple trials to determine the values of the tuning parameters, while VARD requires a single run. In these cases, VARD can still offer a lower *total* run-time than the alternative methods and it may be more convenient since it does not rely on human expert evaluation of the different images for different values of tuning parameters.

Each iteration of SBL involves the solution of several $p \times p$ matrix equations using the conjugate gradient (CG) method [59] in order to estimate the posterior variances (see details in Appendix C) with one additional equation for computing the mean. The solution to each matrix equation requires multiple CG sub-iterations, each sub-iteration requiring one forward projection and one back-projection. Denoting K as the number of matrix equations for the variance estimation, and denoting I_m, I_v as the number of CG sub-iterations for the mean and variance parts, respectively, each iteration of SBL involves $K \times I_v + I_m$ forward/back-projections. The choice for K and the typical values of I_m, I_v depend on the specific system and object of interest. For the example in Sec. 9.1 we used $K = 64$, and the chosen residual thresholds (see Appendix C) lead to $I_v \approx 10$ and $I_m \approx 100$, resulting in 2 orders of magnitude more forward and back-projections per iteration than VARD.

10. Conclusions. We presented a new framework for automatic relevance determination (ARD), which we call variational ARD (VARD). It provides an extension of previous ARD methods to Poisson noise models for transmission tomography and allows the use of neighborhood penalties that promote sparsity in the pixel/voxel difference domain. We revealed the sparsity-promoting mechanism in VARD, and established key differences compared to previous ARD methods that are applicable only to Gaussian noise models. In addition to computing the estimate for the imaged object, which is given by an approximation to the posterior mean, VARD also provides posterior variances, i.e., Bayesian confidence intervals. An important advantage of the VARD framework is that *it avoids the cumbersome process of choosing tuning parameters, while still retaining good image quality*. We have shown that the posterior variances play a key role in learning the correct balance between the datafit and the prior. Straightforward simplifications that avoid using posterior variances, such as reweighted ℓ_2 , do not retain this auto-tuning property.

We presented a parallel AM algorithm for VARD (PAR-VARD) which is applicable to large scale problems. We studied a parallel implementation of VARD using both synthetic and real data. The numerical experiments demonstrate that the images produced by VARD are comparable in quality to the best maximum a posteriori (MAP) and reweighted ℓ_2 estimates.

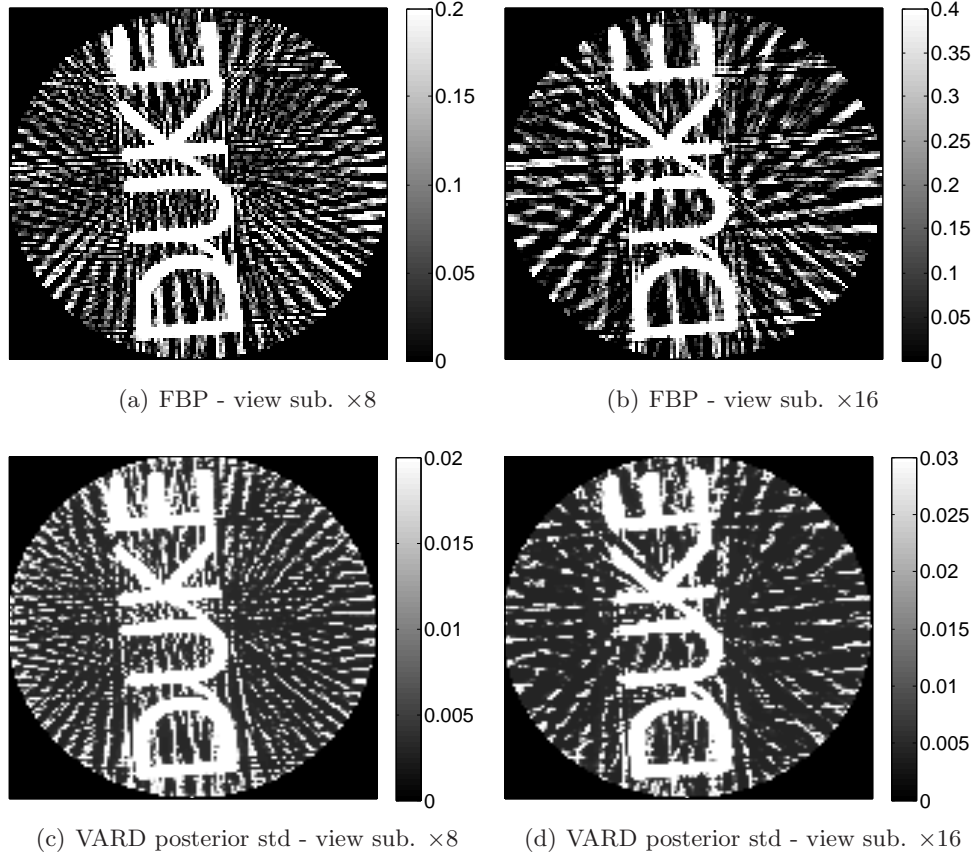


Figure 10. Comparing artifacts in FBP to the posterior variances obtained via VARD. (a) Zoom-in view of Fig. 9(c); (b) Zoom-in view of Fig. 9(d); (c) VARD posterior std using 1/8 of the views; (d) same as (c) but using 1/16 of the views. Note the resemblance between the streaks in (a) and (c), and also (b) and (d).

However, VARD needs to be run only once, whereas MAP and reweighted ℓ_2 need to be run multiple times whenever the tuning parameters are unknown or better ones are needed. For MAP-based algorithms, it is possible to speed-up the search for tuning parameters by employing a continuation approach where the algorithm is initialized using the solution from a previous point in parameter space. A more principled approach to determine the tuning parameters in MAP-based algorithms is cross-validation, which requires multiple runs for different subsets of the data and becomes computationally involved when the number of tuning parameters increases.

The unique features of VARD such as auto-tuning and computation of variances do not come without a price. We have shown that VARD has at least double computational complexity than a MAP-based algorithm for large scale problems. For commercial CT systems which are based on protocols with preset values for the tuning parameters, the auto-tuning feature of VARD might not be needed and would entail a longer reconstruction time. However, we expect our method to be useful in non-standard situations when the preset values for the tuning parameters are not satisfactory and better values need to be found. In addition, auto-tuning

can greatly simplify research when investigating non-standard geometries for which the tuning parameters might not be known. Exploring ways to reduce the computational burden of VARD so that it compares more favorably to MAP-based algorithms seems like an important direction for future work.

Nevertheless, we have considerably advanced the computational feasibility of ARD for large scale problems in this work. In the EM algorithm for ARD [15] with a Gaussian noise model, the straightforward computation of the posterior variances scales cubically with the number of pixels/voxels, which is infeasible for large scale problems, so approximations are needed. The scalable ARD algorithm in [17] (SBL) which uses sophisticated approximations to the variances does not have convergence guarantees. In contrast, the proposed VARD algorithm is feasible for large-scale problems and also has *guarantees for global convergence*. By comparing VARD to SBL, we have shown that for low photon flux (high noise levels) VARD leads to better image quality than SBL, when the data is generated from the considered Poisson model. Furthermore, this is also *true for high photon flux* (low noise levels), when the Gaussian approximation to the Poisson distribution is good enough, and seems to be due to the over-complete prior that can be used by VARD but cannot be used by SBL. We have shown that VARD is much more efficient computationally than SBL, requiring considerably less forward/back-projections (the main computational bottleneck), and VARD is also much simpler to implement. As opposed to previous ARD methods [15, 17], the exact evaluation of the VARD objective is feasible for large-scale problems and can be used to assess convergence in practice and to verify the correctness of the implementation.

We have shown that when the object is undersampled, the posterior variances computed by VARD can indicate regions in the image where sampling is insufficient. The variances can therefore be used to define relative levels of confidence in the reconstruction at particular regions in the image. The variances can also be used for adaptive sensing/experimental design [17, 30]. Our results suggest that the variances can be used for image segmentation or boundary detection. These potential uses are not studied here, and will be the subject of future work.

Finally, we note that many algorithms for transmission tomography use ordered subsets (OS) to accelerate convergences while sacrificing guarantees for convergence [6, 12, 47, 48]. In this paper we have excluded the use of ordered subsets and focused on convergent algorithms in order to be consistent with the convergence theory presented in the paper. Initial experiments with OS for VARD appear to be promising but will be reported in a future publication.

11. Acknowledgments. We would like to thank Prof. Martin P. Tornai for providing us access to the Duke Multi-Modality Imaging Laboratory (MMIL) where the experimental data was acquired. We would like to thank Mr. Andrew Holmgren for conducting the experiments. We thank Dr. Ikenna Odinaka for suggesting the 1D trust-region Newton method used in our code. We thank Dr. David Carlson for proofreading the manuscript and providing helpful comments. This work was supported by the Department of Homeland Security, Science and Technology Directorate, Explosives Division, through contract HSHQDC-11-C-0083.

Appendix A. Proof of Theorem 4.9.

Proof. We split the difference in (4.17) into two terms

$$\begin{aligned} \mathcal{F}(\mathbf{m}^{(t)}, \mathbf{v}^{(t)}, \gamma^{(t)}) - \mathcal{F}(\mathbf{m}^{(t+1)}, \mathbf{v}^{(t+1)}, \gamma^{(t+1)}) &= \mathcal{F}(\mathbf{m}^{(t)}, \mathbf{v}^{(t)}, \gamma^{(t)}) - \mathcal{F}(\mathbf{m}^{(t+1)}, \mathbf{v}^{(t+1)}, \gamma^{(t)}) \\ (A.1) \quad &+ \mathcal{F}(\mathbf{m}^{(t+1)}, \mathbf{v}^{(t+1)}, \gamma^{(t)}) - \mathcal{F}(\mathbf{m}^{(t+1)}, \mathbf{v}^{(t+1)}, \gamma^{(t+1)}). \end{aligned}$$

First we focus on the first difference in (A.1) and write it explicitly using (3.4)-(3.5)

$$\begin{aligned} \mathcal{F}(\mathbf{m}^{(t)}, \mathbf{v}^{(t)}, \gamma^{(t)}) - \mathcal{F}(\mathbf{m}^{(t+1)}, \mathbf{v}^{(t+1)}, \gamma^{(t)}) &= \sum_j b_j^{(t)} (m_j^{(t)} - m_j^{(t+1)}) + \sum_i \mu_i^{(t)} - \\ &\sum_i \mu_i^{(t)} \exp \left[- \sum_j \phi_{ij} (m_j^{(t+1)} - m_j^{(t)}) + \sum_j \phi_{ij}^2 (v_j^{(t+1)} - v_j^{(t)})/2 \right] + \\ &\sum_k \frac{1}{2\gamma_k^{(t)}} [(\sum_j \psi_{kj} m_j^{(t)})^2 - (\sum_j \psi_{kj} m_j^{(t+1)})^2] + \\ (A.2) \quad &\sum_k \frac{1}{2\gamma_k^{(t)}} [\sum_j \psi_{kj}^2 (v_j^{(t)} - v_j^{(t+1)})] + \sum_j \log(v_j^{(t+1)}/v_j^{(t)})/2. \end{aligned}$$

Next, we present several bounds for terms in (A.2). Using (3.7)-(3.8) and (3.5) we obtain

$$(A.3) \quad \sum_i \mu_i^{(t)} \geq \sum_i \mu_i^{(t)} \sum_j \frac{\phi_{ij} + \phi_{ij}^2/2}{Z_1} = \sum_j \frac{b_j^{(t)}}{Z_1} + \frac{\tilde{b}_j^{(t)}}{Z_1} \geq \sum_j \frac{b_j^{(t)}}{Z_j^{(t)}} + \frac{\tilde{b}_j^{(t)}}{Z_1},$$

where the last inequality is based on the definition for $Z_j^{(t)}$ in (4.12). We bound the exponential term in (A.2) using the convex decomposition lemma (lemma 3.1)

$$\begin{aligned} & - \sum_i \mu_i^{(t)} \exp \left(- \sum_j \phi_{ij} (m_j^{(t+1)} - m_j^{(t)}) + \sum_j \phi_{ij}^2 (v_j^{(t+1)} - v_j^{(t)})/2 \right) \geq \\ (A.4) \quad & - \sum_j \frac{b_j^{(t)}}{Z_j^{(t)}} \exp \left(- Z_j^{(t)} (m_j^{(t+1)} - m_j^{(t)}) \right) - \sum_j \frac{\tilde{b}_j^{(t)}}{Z_1} \exp \left(Z_1 (v_j^{(t+1)} - v_j^{(t)}) \right), \end{aligned}$$

where we used (3.8) with Z_1 replaced by $Z_j^{(t)} \geq Z_1$ in r_{ij} (lemma 3.1 with the dummy variable extension applied), and also used (3.5). Using lemma 3.1 for the quadratic term in (A.2) we obtain

$$\begin{aligned} (A.5) \quad & \sum_k \frac{1}{2\gamma_k^{(t)}} [(\sum_j \psi_{kj} m_j^{(t)})^2 - (\sum_j \psi_{kj} m_j^{(t+1)})^2] \geq - \sum_j [f_j^{(t)} (m_j^{(t+1)} - m_j^{(t)}) - g_j^{(t)} (m_j^{(t+1)} - m_j^{(t)})^2], \end{aligned}$$

where the derivation is similar to the one in (3.10) and we also used (3.13)-(3.14). Substituting (4.11), (4.13) into the right hand sides of (A.4) and (A.5) and then substituting the result

into (A.2), we obtain

$$(A.6) \quad \begin{aligned} & \mathcal{F}(\mathbf{m}^{(t)}, \mathbf{v}^{(t)}, \boldsymbol{\gamma}^{(t)}) - \mathcal{F}(\mathbf{m}^{(t+1)}, \mathbf{v}^{(t+1)}, \boldsymbol{\gamma}^{(t)}) \geq \\ & \sum_j \left\{ \frac{b_j^y}{Z_j^{(t)}} \log \left(\frac{\beta_j^{(t)}}{b_j^{(t)}} \right) + \frac{b_j^{(t)}}{Z_j^{(t)}} + \frac{\tilde{b}_j^{(t)}}{Z_1} - \frac{\beta_j^{(t)}}{Z_j^{(t)}} - \frac{\tilde{\beta}_j^{(t)}}{Z_1} + \frac{f_j^{(t)}}{Z_j^{(t)}} \log \left[\frac{\beta_j^{(t)}}{b_j^{(t)}} \right] + \right. \\ & \left. \frac{g_j^{(t)}}{Z_j^{(t)}} \log \left[\frac{\beta_j^{(t)}}{b_j^{(t)}} \right] (m_j^{(t+1)} - m_j^{(t)}) - \frac{\xi_j^{(t)}}{2Z_1} \log \left[\frac{\tilde{\beta}_j^{(t)}}{\tilde{b}_j^{(t)}} \right] + \log(v_j^{(t+1)}/v_j^{(t)})/2 \right\}. \end{aligned}$$

We also use

$$(A.7) \quad \begin{aligned} (m_j^{(t+1)} - m_j^{(t)}) &= 2(\bar{m}_j^{(t)} - m_j^{(t)}) + (m_j^{(t+1)} - m_j^{(t)} + 2(m_j^{(t)} - \bar{m}_j^{(t)})) = \\ & 2(\bar{m}_j^{(t)} - m_j^{(t)}) + \left(\frac{2}{Z_1} - \frac{1}{Z_j^{(t)}} \right) \log \left(\frac{\beta_j^{(t)}}{b_j^{(t)}} \right) \geq 2(\bar{m}_j^{(t)} - m_j^{(t)}) + \frac{1}{Z_j^{(t)}} \log \left(\frac{\beta_j^{(t)}}{b_j^{(t)}} \right), \end{aligned}$$

where we have used (4.7) and (4.11). Next, we use (4.10) in (A.7) to obtain

$$(A.8) \quad (m_j^{(t+1)} - m_j^{(t)}) \geq (\beta_j^{(t)} - b_j^y - f_j^{(t)})/g_j^{(t)} + \frac{1}{Z_j^{(t)}} \log \left(\frac{\beta_j^{(t)}}{b_j^{(t)}} \right)$$

and substituting into (A.6) we have

$$(A.9) \quad \begin{aligned} & \mathcal{F}(\mathbf{m}^{(t)}, \mathbf{v}^{(t)}, \boldsymbol{\gamma}^{(t)}) - \mathcal{F}(\mathbf{m}^{(t+1)}, \mathbf{v}^{(t+1)}, \boldsymbol{\gamma}^{(t)}) \geq \frac{b_j^{(t)}}{Z_j^{(t)}} + \frac{\tilde{b}_j^{(t)}}{Z_1} - \frac{\beta_j^{(t)}}{Z_j^{(t)}} - \frac{\tilde{\beta}_j^{(t)}}{Z_1} + \\ & g_j^{(t)} \left[\frac{1}{Z_j^{(t)}} \log \left(\frac{\beta_j^{(t)}}{b_j^{(t)}} \right) \right]^2 + \frac{\beta_j^{(t)}}{Z_j^{(t)}} \log \left(\frac{\beta_j^{(t)}}{b_j^{(t)}} \right) - \frac{\xi_j^{(t)}}{2Z_1} \log \left(\frac{\tilde{\beta}_j^{(t)}}{\tilde{b}_j^{(t)}} \right) + \frac{1}{2} \log \left(\frac{v_j^{(t+1)}}{v_j^{(t)}} \right). \end{aligned}$$

Finally, we use $\log(x) \geq 1 - 1/x$ applied to the last term in (A.9), which gives

$$(A.10) \quad \log(v_j^{(t+1)}/v_j^{(t)}) \geq \frac{1}{v_j^{(t+1)}}(v_j^{(t+1)} - v_j^{(t)}) = \frac{1}{v_j^{(t+1)}} \frac{1}{Z_1} \log \left(\frac{\tilde{\beta}_j^{(t)}}{\tilde{b}_j^{(t)}} \right),$$

where we used (4.13). Substituting (A.10) into (A.9), using (4.14) and the definition of the I-divergence in (4.15), we obtain

$$(A.11) \quad \mathcal{F}(\mathbf{m}^{(t)}, \mathbf{v}^{(t)}, \boldsymbol{\gamma}^{(t)}) - \mathcal{F}(\mathbf{m}^{(t+1)}, \mathbf{v}^{(t+1)}, \boldsymbol{\gamma}^{(t)}) \geq I \left(\frac{\boldsymbol{\beta}^{(t)}}{\mathbf{Z}^{(t)}} \parallel \frac{\mathbf{b}^{(t)}}{\mathbf{Z}^{(t)}} \right) + I \left(\frac{\tilde{\boldsymbol{\beta}}^{(t)}}{Z_1} \parallel \frac{\tilde{\mathbf{b}}^{(t)}}{Z_1} \right).$$

Next, we calculate the second difference in (A.1)

$$(A.12) \quad \begin{aligned} & \mathcal{F}(\mathbf{m}^{(t+1)}, \mathbf{v}^{(t+1)}, \boldsymbol{\gamma}^{(t)}) - \mathcal{F}(\mathbf{m}^{(t+1)}, \mathbf{v}^{(t+1)}, \boldsymbol{\gamma}^{(t+1)}) = \\ & \frac{1}{2} \sum_k \left(\frac{1}{\gamma_k^{(t)}} - \frac{1}{\gamma_k^{(t+1)}} \right) \left[\left(\sum_j \psi_{kj} m_j^{(t+1)} \right)^2 + \sum_j \psi_{kj}^2 v_j^{(t+1)} \right] + \frac{1}{2} \sum_k \log \left(\frac{\gamma_k^{(t)}}{\gamma_k^{(t+1)}} \right). \end{aligned}$$

Substituting (2.16) into (A.12) we obtain

$$(A.13) \quad \mathcal{F}(\mathbf{m}^{(t+1)}, \mathbf{v}^{(t+1)}, \gamma^{(t)}) - \mathcal{F}(\mathbf{m}^{(t+1)}, \mathbf{v}^{(t+1)}, \gamma^{(t+1)}) = \frac{1}{2} \sum_k \log \left(\frac{\gamma_k^{(t)}}{\gamma_k^{(t+1)}} \right) + \frac{1}{2} \left(\frac{\gamma_k^{(t+1)}}{\gamma_k^{(t)}} - 1 \right) \triangleq \frac{1}{2} D_{IS}(\gamma^{(t)} \parallel \gamma^{(t+1)}),$$

where we used the definition in (4.16). Combining the results in (A.11) and (A.13) we have

$$(A.14) \quad \begin{aligned} \mathcal{F}(\mathbf{m}^{(t)}, \mathbf{v}^{(t)}, \gamma^{(t)}) - \mathcal{F}(\mathbf{m}^{(t+1)}, \mathbf{v}^{(t+1)}, \gamma^{(t+1)}) &= \mathcal{F}(\mathbf{m}^{(t)}, \mathbf{v}^{(t)}, \gamma^{(t)}) - \mathcal{F}(\mathbf{m}^{(t+1)}, \mathbf{v}^{(t+1)}, \gamma^{(t)}) + \\ &\mathcal{F}(\mathbf{m}^{(t+1)}, \mathbf{v}^{(t+1)}, \gamma^{(t)}) - \mathcal{F}(\mathbf{m}^{(t+1)}, \mathbf{v}^{(t+1)}, \gamma^{(t+1)}) \geq I \left(\frac{\beta^{(t)}}{\mathbf{Z}^{(t)}} \parallel \frac{\mathbf{b}^{(t)}}{\mathbf{Z}^{(t)}} \right) + I \left(\frac{\tilde{\beta}^{(t)}}{\mathbf{Z}_1} \parallel \frac{\tilde{\mathbf{b}}^{(t)}}{\mathbf{Z}_1} \right) + \\ &\sum_j g_j^{(t)} \left[\frac{1}{Z_j^{(t)}} \log \left(\frac{\beta_j^{(t)}}{b_j^{(t)}} \right) \right]^2 + \frac{1}{2} D_{IS}(\gamma^{(t)} \parallel \gamma^{(t+1)}) \end{aligned}$$

which is the result stated in Theorem 4.9. ■

Appendix B. Implementation Details. Next, we address the implementation of lines 11-12 in algorithm 1 which involve minimization of 1D surrogate functions. Since it is not necessary to find the exact solutions for the surrogate problems, we use Newton's method with a single Newton step when solving for the mean and using thresholding to impose positivity, i.e.,

$$(B.1) \quad m_j^{(t+1)} = [m_j^{(t)} - (\partial_x \mathcal{S}(x) / \partial_x^2 \mathcal{S}(x))|_{x=m_j^{(t)}}]_+ = [m_j^{(t)} - (b_j^y - b_j^{(t)} + f_j^{(t)}) / (Z_1 b_j^{(t)} + 2g_j^{(t)})]_+,$$

where ∂_x denotes the derivative with respect to x , \mathcal{S} denotes the surrogate in (3.17), and all parameters are defined in (3.5), (3.8), (3.13), (3.14). While Newton steps as in (B.1) are not guaranteed to decrease the objective function, we have found that in all our numerical experiments the objective function decreased at each iteration. If the objective function is not monotonically decreasing, then the step size can be adjusted or a trust-region approach can be used, as described next for the variance updates.

When solving for the variance, we use a Newton trust region approach [61] where Newton steps for v_j are considered (similarly to (B.1)) but the step sizes are adjusted such that the solution will lie within a trusted region. At each iteration, the trust region is adjusted from both left and right by comparing predicted reduction based on a local quadratic approximation and the actual reduction of the 1D surrogate function. If the prediction is poor, the trust region is shrunk. The left boundary is limited to the positive axis. This approach also ensures the decrease of the objective function at each step.

For MAP with the smooth edge-preserving neighborhood penalties we found that a trust region approach is also required to guarantee convergence in cases where the linear part of the penalty dominates. For reweighted ℓ_2 , taking a single Newton step each iteration was sufficient.

Appendix C. Details about the SBL Algorithm. We provide a short overview of the SBL algorithm in [17]. For further details please see [15, 17]. A commonly used alternative to the Poisson noise model in (1.1) is the post-log Gaussian noise model given by

$$(C.1) \quad \tilde{y}_i = \log(\eta_i/y_i), \quad p(\tilde{\mathbf{y}}|\mathbf{x}) = \prod_{i=1}^n p(\tilde{y}_i|\mathbf{x}) = \prod_{i=1}^n \mathcal{N}(\tilde{y}_i|\phi_i^T \mathbf{x}, y_i^{-1}),$$

where $\tilde{\mathbf{y}}$ denotes the post-log calibrated measurements. Note that the variance is inversely proportional to the data y_i . The Gaussian model in (C.1) can approximate the Poisson model in (1.1) quite well for moderate to high photon flux. As described in Sec. 1.2.2, the solution to (1.7) under the Gaussian model in (C.1) can be found using an expectation maximization (EM) algorithm [15]. In the expectation (E) step, one uses the conjugate gradient method [59] to solve the following matrix equation for the posterior mean \mathbf{m} [17]

$$(C.2) \quad \mathbf{P}\mathbf{m} = \Phi^T \mathbf{B}\tilde{\mathbf{y}},$$

where \mathbf{P} is a $p \times p$ matrix, defined as

$$(C.3) \quad \mathbf{P} = \Phi^T \mathbf{B}\Phi + \Psi^T \mathbf{\Gamma}^{-1} \Psi,$$

with $\mathbf{B} = \text{Diag}(\mathbf{y})$ and $\mathbf{\Gamma} = \text{Diag}(\boldsymbol{\gamma})$ ($\boldsymbol{\gamma}$ and Ψ are defined in (2.1)). Another operation required for the E-step is the computation of the diagonal of the covariance matrix [17]

$$(C.4) \quad \boldsymbol{\Sigma} = \Psi^T \mathbf{P}^{-1} \Psi,$$

which direct computation has $O(p^3)$ complexity (p =number of pixels/voxels) due to the matrix inverse, which is not practical. To resolve this issue, Jeon et al. [17] used an estimator for the diagonal of the covariance matrix based on a method proposed by Tang and Saad [60]. The general idea is to use the conjugate gradient (CG) method [59] to solve the following matrix equations for \mathbf{x}_k

$$(C.5) \quad \mathbf{P}\mathbf{x}_k = \Psi^T \mathbf{q}_k, \quad k = 1, 2, \dots, K$$

where \mathbf{P} is given in (C.3) and \mathbf{q}_k are user-specified “probing vectors”, and then use all the solutions \mathbf{x}_k ($k = 1, 2, \dots, K$) to construct an estimator for the diagonal of $\boldsymbol{\Sigma}$ in (C.4) in a manner described in [17]. Since the CG method only requires the multiplication of the matrix \mathbf{P} with some vector, the computational complexity is reduced from $O(p^3)$ to $O(np^{1/D})$ ($D = 2, 3$ for 2D/3D images, respectively). Here we used the probing vectors that are based on the graph coloring scheme in [17]. Note that the structure and number of probing vectors required for some given accuracy depend on the properties of the matrix \mathbf{P} , so they are system and object dependent. In the M-step, $\boldsymbol{\gamma}$ is updated as

$$(C.6) \quad \boldsymbol{\gamma} = [(\Psi\mathbf{m}) \odot (\Psi\mathbf{m}) + \text{diag}(\boldsymbol{\Sigma})]^{-1},$$

where the inverse is performed elementwise. The E and M steps are repeated till convergence.

We define the residuals for the matrix equations in (C.2) and (C.5) as $R_m = \|\mathbf{P}\mathbf{m} - \Phi^T \mathbf{B}\tilde{\mathbf{y}}\|_2$ and $R_v = \|\mathbf{P}\mathbf{x}_k - \Psi^T \mathbf{q}_k\|_2$, respectively, for a given \mathbf{m} and \mathbf{x}_k ($k = 1, 2, \dots, K$).

The CG iterations for (C.2) and (C.5) are stopped when $R_m \leq \varepsilon_m$ and $R_v \leq \varepsilon_v$, where ε_m and ε_v are thresholds for the residuals. In the example of Sec. 9.1, we performed several trials with different choices for K and for the thresholds ε_m and ε_v . Based on the outcomes of these trials, we chose $K = 64$, $\varepsilon_m = 10^{-8}$, and $\varepsilon_v = 10^{-2}$. Note that decreasing ε_v had negligible effect on the accuracy of the mean \mathbf{m} . Increasing ε_m led to significantly larger error in \mathbf{m} in later iterations.

Appendix D. Details for the Experimental Lab Setup. The real data has been acquired at the Duke Multi-Modality Imaging Lab (MMIL) [62]. The x-ray source specifications are 60kVp, 50mA, and 25ms. A 0.55mm Cerium filter was used to strongly filter the Bremsstrahlung (continuous spectrum x-rays) and create a pseudo monoenergetic source spectrum. The detector has a pixel pitch of $127\mu\text{m}$ and a fill-factor of 0.75 (pixel pitch divided by pixel size). By integrating the flux-energy curve computed using the XSPECT software [63] and accounting for detector fill-factor, we estimated a total of $\eta = 2 \times 10^3$ photons per detector pixel. The view angle resolution is 1° .

REFERENCES

- [1] A. C. Kak and M. Slaney, *Principles of Computerized Tomographic Imaging*, SIAM, 2001, <http://www.slaney.org/pct/>.
- [2] F. Natterer, *The Mathematics of Computerized Tomography*, John Wiley & Sons, 1986.
- [3] W. A. Kalender, *X-ray computed tomography*, Phys. Med. Biol., 51(13), 2006, pp. R29-R43.
- [4] J. Frank, *Electron tomography*, Springer, 1992.
- [5] M. H. Li, H. Q. Yang, and H. Kudo, *An accurate iterative reconstruction algorithm for sparse objects: Application to 3d blood vessel reconstruction from a limited number of projections*, Phys. Med. Biol., 47, 2002, pp. 2599–2609.
- [6] J. A. Fessler, *Handbook of Medical Imaging Vol. 2, Chapter 1: Statistical Image Reconstruction Methods for Transmission Tomography*, Edited by M. Sonka and J. M. Fitzpatrick, SPIE, 2000.
- [7] E. P. Simoncelli, *Modeling the joint statistics of images in the wavelet domain*, Proc. SPIE 1999, pp. 188–195.
- [8] L. Boubchir and J. M. Fadili, *Multivariate statistical modeling of images with the curvelet transform*, Proc. 8th International Conference on Signal Proc. and Its Applications, 2005, pp. 747–750.
- [9] K. Lang and R. Carson, *EM reconstruction algorithms for emission and transmission tomography*, J. Comput. Assist. Tomog., 8(2), 1984, pp. 306–316.
- [10] P. W. Holland, and R. E. Welsch, *Robust regression using iteratively reweighted least-squares*, Communications in Statistics-theory and Methods, 6(9), pp. 813–827, 1977.
- [11] K. Lange and J. A. Fessler, *Globally convergent algorithms for maximum a posteriori transmission tomography*, IEEE Trans. Image Proc., 4(10), 1995, pp. 1430–1438.
- [12] H. Erdogan, J.A. Fessler, *Ordered subsets algorithms for transmission tomography*, Phys. Med. Biol., vol. 44, pp. 1999, pp. 2835–2851.
- [13] J. A. O’Sullivan and J. Benac, *Alternating minimization algorithms for transmission tomography*, IEEE Trans. Med. Imaging, 26(3), 2007, pp. 283–297.
- [14] R. M. Neal, *Bayesian Learning for Neural Networks*, Springer-Verlag, 1996.
- [15] M. E. Tipping, *Sparse Bayesian learning and the relevance vector machine*, J. Mach. Learn. Res., 1, 2001, pp. 211–244.
- [16] D. P. Wipf, B. D. Rao, and S. Nagarajan, *Latent variable Bayesian models for promoting sparsity*, IEEE Trans. Inform. Theory, 57, 2011, pp. 6236–6255.
- [17] H. Jeon, Y. Kaganovsky, S. Han, and L. Carin, *GPU-based sparse Bayesian learning for adaptive transmission tomography*, Nuclear Science Symposium and Medical Imaging Conference, IEEE, 2015.
- [18] J. S. Iwanczyk, E. Nygård, O. Meirav, J. Arenson, W. C Barber, N. E. Hartsough, N. Malakhov, and J.

- C Wessel, *Photon counting energy dispersive detector arrays for x-ray imaging*, Nuclear Science, IEEE Transactions on, 56(3), pp 535–542, 2009
- [19] C. A. Bouman, K. Sauer, *A unified approach to statistical tomography using coordinate descent optimization*, IEEE Trans. Image Proc., 5(3), 1996, pp. 480–492.
- [20] S. Ramani, and J. Fessler, *A splitting-based iterative algorithm for accelerated statistical X-ray CT reconstruction*, Medical Imaging, IEEE Transactions on, 31(3), pp. 677–688, 2012.
- [21] P. Sukovic, and N. H. Clinthorne, *Penalized weighted least-squares image reconstruction for dual energy X-ray transmission tomography*, Medical Imaging, IEEE Transactions on, 19(11), pp. 1075–1081, 2000.
- [22] J. Fessler, *Hybrid Poisson/polynomial objective functions for tomographic image reconstruction from transmission scans*, Image Processing, IEEE Transactions on, 4(10), pp. 1439–1450, 1995.
- [23] W. Zhuang, S. S. Gopal, and T. J. Hebert, *Numerical evaluation of methods for computing tomographic projections*, IEEE Trans. Nucl. Sci., 4, 1994, pp. 1660–1665.
- [24] I. Csiszár and G. Tusnády, *Information geometry and alternating minimization procedures*, Statistics and Decisions, Supplementary Issue 1, 1984, pp. 205–237.
- [25] E. Y. Sidky, C. M. Kao, and X. H. Pan, *Accurate image reconstruction from few-views and limited-angle data in divergent-beam CT*, J. X-ray Sci. Tech., 14, 2006, pp. 119–139.
- [26] E. Y. Sidky and X. C. Pan, *Image reconstruction in circular cone-beam computed tomography by constrained total-variation minimization*, Phys. Med. Biol., 53, 2008, pp. 4777–4807.
- [27] D. P. Wipf, J. Palmer, and B. Rao, *Perspectives on sparse Bayesian learning*, Proc. Neural Information Processing Systems, 16(1), 2004, pp. 249–256.
- [28] D. P. Wipf and S. Nagarajan, *A new view of automatic relevance determination*, Proc. Neural Information Processing Systems, 20, 2008.
- [29] A. P. Dempster, N. M. Laird, and D. B. Rubin, *Maximum likelihood from incomplete data via the EM algorithm*, J. Roy. Stat. Soc., 39(1), 1977, pp. 1–38.
- [30] M. Seeger and H. Nickisch, *Large scale Bayesian inference and experimental design for sparse linear models*, SIAM J. Imaging Sci., 4(1), 2011, pp. 166–199.
- [31] M. Seeger and P. Wipf, *Variational Bayesian inference techniques*, IEEE Mag. Signal Proc., 27(6), 2010, pp. 81–91.
- [32] C. Bekas, E. Kokipoulou, and Y. Saad, *An estimator for the diagonal of a matrix*, Applied Numerical Mathematics, 57(11), 2007, pp. 1214–1229.
- [33] G. Papandreou and A. L. Yuille, *Efficient variational inference in large-scale Bayesian compressed sensing*, Proc. IEEE Workshop on Information Theory in Computer Vision and Pattern Recognition, 2011.
- [34] D. G. Luenberger, Y. Ye, *Linear and Nonlinear Programming*, 3rd edition, 2008. Springer,
- [35] R. M. Neal and G. E. Hinton, *A view of the EM algorithm that justifies incremental, sparse, and other variants*, Learning in Graphical Models, Springer, 1998, pp. 355–368.
- [36] A. Gunawardana and W. Byrne, *Convergence theorems for generalized alternating minimization procedures*, J. Mach. Learn. Res., 6, 2005, pp. 2049–2073.
- [37] D. G. Tzikas, A. C. Likas, and N. P. Galatsanos, *The variational approximation for Bayesian inference*, IEEE Mag. Signal Proc., 25(6), 2008, pp. 131–146.
- [38] J. Palmer, D. Wipf, K. Kreutz-Delgado, and B. Rao, *Variational EM algorithms for non-Gaussian latent variable models*, Proc. Advances in Neural Information Processing systems, 18, 2006, page 1059.
- [39] M. Jordan, Z. Ghahramani, T. Jaakkola, and L. Saul, *Introduction to variational methods for graphical models*, Machine Learning, 37, 1999, pp. 183–233.
- [40] C. F. Jeff Wu, *On the convergence properties of the EM algorithm*, The Annals of Statistics, 11, 1983, pp. 95–103.
- [41] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, 2004.
- [42] C. M. Bishop, *Pattern Recognition and Machine Learning*, 2006, Springer New York.
- [43] W. I. Zangwill, *Nonlinear Programming: A Unified Approach*, Prentice-Hall, 1969.
- [44] R. Chartrand and W. Yin, *Iteratively reweighted algorithms for compressive sensing*, Proc. IEEE international conference on Acoustics, speech and signal processing, 2008, pp. 3869–3872.
- [45] D. Wipf and S. Nagarajan, *Iterative reweighted ℓ_1 and ℓ_2 methods for finding sparse solutions*, IEEE Journal of Selected Topics in Signal Proc., 4(2), 2010, pp. 317–329.
- [46] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, 2012, John Wiley & Sons.
- [47] D. Keesing, *Development and Implementation of Fully 3D Statistical Image Reconstruction Algorithms for*

- Helical CT and Half- Ring PET Insert System*, PhD thesis, Washington University in St. Louis, 2009.
- [48] S. Degirmenci, D. G. Politte, C. Bosch, N. Tricha and J. A. O’Sullivan, *Acceleration of iterative image reconstruction for x-ray imaging for security applications*, Proc. SPIE 9401, Computational Imaging XIII, 94010C, March 2015, doi:10.1117/12.2082966.
 - [49] E. Challis and D. Barber, *Concave Gaussian variational approximations for inference in large-scale Bayesian linear models*, Proc. International Conference on Artificial Intelligence and Statistics, 6, 2011, page 7.
 - [50] M. E. Khan, A. Aravkin, M. Friedlander, and M. Seeger, *Fast dual variational inference for non-conjugate latent Gaussian models*, Proc. International Conference on Machine Learning, 2013, pp. 951–959.
 - [51] S. Banerjee, B. P. Carlin, and A. E. Gelfand, *Hierarchical Modeling and Analysis for Spatial Data*, CRC Press, 2014.
 - [52] B. R. Whiting, *Signal Statistics of X-ray Computed Tomography*, Proc. SPIE Conference on Medical Imaging 2002: Physics of Medical Imaging, 4682 (L. Antonuk and M. Yaffe, Eds.), San Diego, CA, Feb. 2002.
 - [53] B. R. Whiting, P. Massoumzadeh, O. A. Earl, J. A. OSullivan, D. L. Snyder, and J. F. Williamson, *X-ray Computed Tomography Signal Properties*, Med. Phys. 33, 3290, 2006.
 - [54] J. Nuyts, B. De Man, J. A. Fessler, W. Zbijewski, and F. J. Beekman, *Modelling the physics in the iterative reconstruction for transmission computed tomography*, Phys. Med. Biol., 58(12), 2013, pp. R63–R96.
 - [55] S. Basu, and B. De Man, *Branchless distance-driven projection and backprojection*, Proc. SPIE Electronic Imaging, 6065, 2006.
 - [56] B. De Man, and S. Basu, *Distance-driven projection and backprojection in three dimensions*, Phys. Med. Biol. 49, 2004, pp. 2463–2475.
 - [57] Z. Yu, J. B. Thibault, C.A. Bouman, K.D. Sauer, J. Hsieh, *Fast Model-Based X-Ray CT Reconstruction Using Spatially Nonhomogeneous ICD Optimization*, IEEE Trans. Image Process., 20(1), 2011, pp. 161–175.
 - [58] <http://www.yan-kaganovsky.com/#!code/c24bp>
 - [59] Y. Saad, *Iterative methods for sparse linear systems*, SIAM, 2003.
 - [60] T. Tang, and Y. Saad, *A Probing method for computing the diagonal of a matrix inverse*, Numerical Linear Algebra with Applications, 19(3), pp. 485–501, 2012.
 - [61] D. C. Sorensen, *Newton’s method with a model trust region modification*, SIAM J. Numerical Analysis, 19 (2), 1982, pp. 409–426.
 - [62] D. J. Crotty, R. L. McKinley, and M. P. Tornai, *Experimental spectral measurements of heavy k-edge filtered beams for x-ray computed mamotomography*, Phys. Med. Biol. 52, 2007, pp. 603.
 - [63] C. W. Dodge, *A rapid method for the simulation of filtered X-ray spectra in diagnostic imaging systems* (ProQuest, 2008).